

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-302175

(43) Date of publication of application : 14.11.1995

(51)Int.Cl.

G06F 3/08

G06F 3/06

G06F 12/16

G11C 16/06

(21)Application number : 06-095125

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 09.05.1994

(72)Inventor : SUKEGAWA HIROSHI

MAKI YASUNORI

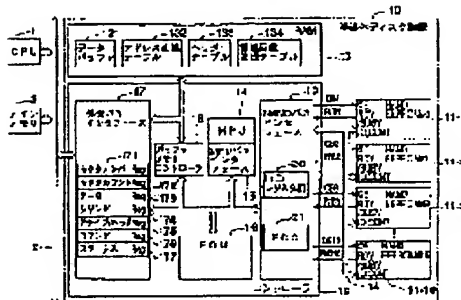
INAGAKI TAKASHI

(54) SEMICONDUCTOR DISK DEVICE

(57)Abstract:

PURPOSE: To improve the reliability of storage data in a flush EEPROM.

CONSTITUTION: An ECC arithmetic circuit 21 calculates data strings transferred from a data buffer 131 in the unit of 256 bytes from a processor 14 to generate an ECC corresponding to the data string. The generated ECC is added to the data string in 256 bytes and transferred to a data register of flush EEPROMs 11-1-11-16. Thus, even when a defective cell of the same bit location of plural pages of the flush EEPROMs takes place, the defective cell including the data string being an object of ECC operation is only one. Thus, error check and correction are implemented by the usual simple ECC arithmetic operation without using a complicated ECC arithmetic equation having a high data recovery capability and then the semiconductor disk device 10 with high storage reliability and capable of high speed access is realized.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision]

of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C): 1998,2003 Japan Patent Office

(11)特許出願公開番号

(43)公開日 平成7年(1995)11月14日

技術表示箇所

3 2 0 A 7608-5B

G 1 1 C 16/06

G 1 1 C 17/ 00

309 F

審査請求 未請求 請求項の数13 O.L (全 31 頁)

特願平6-95125

(71)出題人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 助川 博

東京都青梅市末広町2丁目9番地 株式会社東芝青梅工場内

(72)發明者 牧 康典

東京都青梅市末広町2丁目9番地 株式会社東芝青梅工場内

(72)発明者 稲垣 孝史

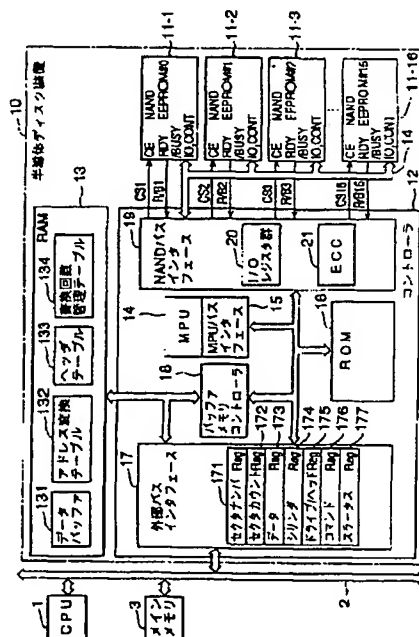
東京都青梅市末広町2丁目9番地 株式会社東芝青梅工場内

(74) 代理人 弁理士 鈴江 武彦

(57)【要約】

【目的】フラッシュEEPROMの記憶データの信頼性の向上を図る。

【構成】ECC演算回路21は、プロセッサ14によってデータバッファ131から256バイト単位で転送されるデータ列を演算し、そのデータ列に対応するECCを生成する。256バイトのデータ列には生成されたECCが付加されて、フラッシュEEPROMのデータレジスタ112に転送される。このため、たとえフラッシュEEPROMの複数ページそれぞれの同一ビット位置に不良セルが発生しても、ECC演算の対象となるデータ列に含まれる不良セルは1つだけとなる。したがって、データ回復能力の高い複雑なECC演算式を利用することなく通常の簡単なECC演算によってエラー検出および訂正を行うことが可能となり、記憶信頼性が高く、且つ高速アクセスが可能な半導体ディスク装置10が実現される。



【特許請求の範囲】

【請求項1】 データ記憶領域および冗長領域を各々が含む複数のページを有するメモリセルアレイと、1ページ分のデータを保持するデータレジスタとを有し、データレジスタとメモリセルアレイ間のデータ転送がページ単位で実行されるフラッシュEEPROMを内蔵し、ホスト装置からのディスクアクセス要求に応じて前記フラッシュEEPROMをアクセスする半導体ディスク装置において

前記ホスト装置から転送されるライトデータおよび前記フラッシュEEPROMから読み出されたリードデータを格納するデータバッファと、

前記ホスト装置からのライト要求に応答して、前記フラッシュEEPROMをライトアクセスするライトアクセス手段とを具備し、

このライトアクセス手段は、

前記データバッファに格納されているライトデータを前記各ページのデータ記憶領域のサイズに相当するデータ列単位で演算して、データ列毎に誤り訂正符号を生成する誤り訂正符号生成手段と、

前記データ列およびそれに対応する誤り訂正符号が同一ページのデータ記憶領域および冗長領域に書き込まれるように、データ列毎にそれに対応する誤り訂正符号を付加して前記フラッシュEEPROMのデータレジスタに転送する手段とを含むことを特徴とする半導体ディスク装置。

【請求項2】 前記ホスト装置からのリード要求に応答して、前記フラッシュEEPROMをリードアクセスするリードアクセス手段をさらに具備し、

このリードアクセス手段は、

前記フラッシュEEPROMから読み出される各ページの誤り訂正符号を演算して、各ページのデータ列毎に誤り検出および訂正を実行する誤り訂正符号演算手段を含むことを特徴とする請求項1記載の半導体ディスク装置。

【請求項3】 前記ホスト装置から指定される不良セクタを示す不良セクタ情報を、その不良セクタに対応するページの冗長領域に書き込む手段をさらに具備することを特徴とする請求項1記載の半導体ディスク装置。

【請求項4】 前記ホスト装置から指定されるECCエラー発生セクタを示すエラー情報を、そのエラー発生セクタに対応するページの冗長領域に書き込む手段をさらに具備することを特徴とする請求項1記載の半導体ディスク装置。

【請求項5】 前記ホスト装置によって実行されるプログラムが格納されたROMをさらに具備し、

前記ホスト装置からの所定番地のリード要求に応答して前記ROMから前記ホスト装置にプログラムが転送されるように構成されていることを特徴とする請求項1記載の半導体ディスク装置。

【請求項6】 データ記憶領域および冗長領域を各々が含む複数のページを有するメモリセルアレイと、1ページ分のデータを保持するデータレジスタとを有し、データレジスタとメモリセルアレイ間のデータ転送がページ単位で実行されるフラッシュEEPROMを内蔵し、ホスト装置からのディスクアクセス要求に応じて前記フラッシュEEPROMをアクセスする半導体ディスク装置において、

マイクロプロセッサと、

10 前記ホスト装置から転送されるライトデータおよび前記フラッシュEEPROMから読み出されたリードデータを格納するデータバッファと、

前記マイクロプロセッサによって制御され、前記フラッシュEEPROMをリード/ライトアクセスするフラッシュメモリインタフェース回路と、

前記マイクロプロセッサによってリード/ライト可能な1/Oレジスタ群とを具備し、

前記フラッシュメモリインタフェース回路は、

20 前記マイクロプロセッサによって前記データバッファから前記各ページのデータ記憶領域のサイズに相当するデータ列単位で読み出されるライトデータを演算して、データ列毎に誤り訂正符号を生成する誤り訂正符号生成回路と、

前記データ列およびそれに対応する誤り訂正符号が同一ページのデータ記憶領域および冗長領域に書き込まれるように、データ列毎にそれに対応する誤り訂正符号を付加して前記フラッシュEEPROMのデータレジスタに転送するライトデータ転送回路と、

30 前記フラッシュEEPROMのデータレジスタから読み出されるリードデータから前記データ記憶領域に格納されていたデータ列を取り出して前記データバッファに転送するリードデータ転送回路と、

前記リードデータに含まれる誤り訂正符号を利用して前記リードデータの誤り検出を実行し、その検出結果を前記1/Oレジスタにセットする誤り訂正符号チェック回路とを含む、

前記マイクロプロセッサは、

前記1/Oレジスタにセットされた演算結果に従って、前記データバッファに転送されたリードデータの誤り訂正を実行することを特徴とする半導体ディスク装置。

【請求項7】 データ記憶領域および冗長領域を各々が含む複数のページをそれぞれ有する複数のブロックを含み、各ブロックの所定ページの冗長領域にそのブロックの書換回数を示す書換回数情報が格納されるメモリセルアレイと、1ページ分のデータを保持するデータレジスタとを有し、データレジスタとメモリセルアレイ間のデータ転送がページ単位で実行され、消去動作がブロック単位で実行されるフラッシュEEPROMを内蔵し、ホスト装置からのディスクアクセス要求に応じて前記フラッシュEEPROMをアクセスする半導体ディスク装置

であって

前記ホスト装置から転送されるライトデータおよび前記フラッシュEEPROMから読み出されたリードデータを格納するデータバッファと、

前記ホスト装置からのライト要求にตอบสนองして、前記フラッシュEEPROMをライトアクセスするライトアクセス手段とを具備し、

このライトアクセス手段は、

前記ライト要求によって指定されるライト対象ブロック内の前記所定ページから書換回数情報を読み出し、その書換回数情報の値を更新する手段と、

前記所定ページに書き込むべきライトデータに前記更新された書換回数情報を付加してページデータを生成し、前記ライトデータおよび前記更新された書換回数情報が前記所定ページのデータ記憶領域および冗長領域に格納されるように、前記ページデータを前記フラッシュEEPROMのデータレジスタに転送する手段とを具備することを特徴とする半導体ディスク装置。

【請求項8】 データ記憶領域および冗長領域を各々が含む複数のページをそれぞれ有する複数のブロックを含み、各ブロックの所定ページの冗長領域にそのブロックの書換回数を示す書換回数情報が格納されるメモリセルアレイと、1ページ分のデータを保持するデータレジスタとを有し、外部からのコマンドにตอบสนองして、ブロック単位で記憶内容を消去する消去動作、メモリセルアレイにページ単位でデータを書込むライト動作、およびメモリセルアレイからページ単位でデータを読み出すリード動作を実行するフラッシュEEPROMを内蔵し、ホスト装置からのディスクアクセス要求に応じて前記フラッシュEEPROMをアクセスする半導体ディスク装置において、

前記ホスト装置から転送されるライトデータおよび前記フラッシュEEPROMから読み出されたリードデータを格納するデータバッファと、

前記ホスト装置からのライト要求にตอบสนองして、前記フラッシュEEPROMをライトアクセスするライトアクセス手段とを具備し、

このライトアクセス手段は、

リードコマンドを発行して前記フラッシュEEPROMをリード動作モードに設定し、前記ライト要求によって指定されたライト対象ブロック内の前記所定ページから書換回数情報を読み出し、その書換回数情報の値を更新する手段と、

ライト対象ブロック内にライトアクセス対象外のページが存在する際、そのライトアクセス対象外のページに格納されているデータを前記データバッファ上に読み出し、その読み出したデータのデータバッファ上のアドレスと前記データバッファに格納されているライトデータのアドレスとに基づいて前記ライト対象のブロックに書き込むべき1ブロック分のブロックデータを構成するデ

ータそれぞれのデータバッファ上の格納位置を示すアドレス情報を保持するヘッダテーブルを生成する手段と、イレースコマンドを発行して前記フラッシュEEPROMを消去動作モードに設定し、前記ライト対象ブロックの記憶内容を消去する手段と、

前記ヘッダテーブルのアドレス情報を参照して、前記データバッファから前記所定ページに書き込むべきライトデータを読み出し、そのライトデータに前記更新された書換回数情報を付加してページデータを生成し、前記ライトデータおよび前記更新された書換回数情報が前記所定ページのデータ記憶領域および冗長領域に格納されるように、前記ページデータを前記フラッシュEEPROMのデータレジスタに転送する手段と、

前記ヘッダテーブルのアドレス情報を参照して、前記データバッファから前記所定ページ以外に書き込むべきライトデータを順次読み出し、それを前記フラッシュEEPROMのデータレジスタに転送する手段とを含むことを特徴とする半導体ディスク装置。

【請求項9】 データ記憶領域および冗長領域を各々が含む複数のページをそれぞれ有する複数のブロックを含み、各ブロックの所定ページの冗長領域にそのブロックの書換回数を示す書換回数情報が格納されるメモリセルアレイと、1ページ分のデータを保持するデータレジスタとを有し、外部からのコマンドにตอบสนองして、ブロック単位で記憶内容を消去する消去動作、メモリセルアレイにページ単位でデータを書込むライト動作、およびメモリセルアレイからページ単位でデータを読み出すリード動作を実行するフラッシュEEPROMを内蔵し、ホスト装置からのディスクアクセス要求に応じて前記フラッシュEEPROMをアクセスする半導体ディスク装置において、

前記ホスト装置から転送されるライトデータおよび前記フラッシュEEPROMから読み出されたリードデータを格納するデータバッファと、

前記ホスト装置からのライト要求にตอบสนองして、前記フラッシュEEPROMをライトアクセスするライトアクセス手段とを具備し、

このライトアクセス手段は、

リードコマンドを発行して前記フラッシュEEPROMをリード動作モードに設定し、前記ライト要求によって指定されたライト対象ブロック内の前記所定ページから書換回数情報を読み出し、その書換回数情報の値を更新する手段と、

ライト対象ブロック内にライトアクセス対象外のページが存在する際、そのライトアクセス対象外のページに格納されているデータを前記データバッファ上に読み出し、その読み出したデータの前記データバッファ上のアドレスと前記データバッファに格納されているライトデータのアドレスとに基づいて前記ライト対象のブロックに書き込むべき1ブロック分の

ブロックデータを構成するデータそれぞれのデータバッファ上の格納位置を示すアドレス情報を保持するヘッダテーブルを生成する手段と、

イレースコマンドを発行して前記フラッシュEEPROMを消去動作モードに設定し、前記ライト対象ブロックの記憶内容を消去する手段と、

前記ヘッダテーブルのアドレス情報を参照して、前記データバッファから前記ブロックデータを前記データ記憶領域のサイズに相当するデータ列単位で順次読み出す手段と

前記データバッファから読み出されるデータ列毎に誤り訂正符号を生成する誤り訂正符号生成手段と、

前記所定ページに書き込むべきデータ列にそのデータ列に対応する誤り訂正符号、および前記更新された書換回数情報を付加してページデータを生成し、前記データ列が前記所定ページのデータ記憶領域、前記更新された書換回数情報および誤り訂正符号が前記所定ページの冗長領域に書き込まれるように、前記生成したページデータを前記フラッシュEEPROMのデータレジスタに転送する手段と、

前記所定ページ以外の他のページ毎に、そのページに書き込むべきデータ列にそのデータ列に対応する誤り訂正符号を付加してページデータを生成し、前記データ列および誤り訂正符号が各ページのデータ記憶領域および冗長領域に書き込まれるように、前記生成したページデータを前記フラッシュEEPROMのデータレジスタに転送する手段とを含むことを特徴とする半導体ディスク装置。

【請求項10】 データ記憶領域および冗長領域を各々を含む複数のページをそれぞれ有する複数のブロックを含み、各ブロックの書換回数を示す書換回数データの上位ビット部から構成される第1の書換回数データが所定ブロックに格納され、各ブロックの所定ページの冗長領域にそのブロックに対応する書換回数データの下位ビット部から構成される第2の書換回数データが格納されるメモリアルレイと、1ページ分のデータを保持するデータレジスタとを有し、外部からのコマンドにตอบสนองして、ブロック単位で記憶内容を消去する消去動作、メモリアルレイにページ単位でデータを書込むライト動作、およびメモリアルレイからページ単位でデータを読み出すリード動作を実行するフラッシュEEPROMを内蔵し、ホスト装置からのディスクアクセス要求に応じて前記フラッシュEEPROMをアクセスする半導体ディスク装置において、

前記ホスト装置から転送されるライトデータおよび前記フラッシュEEPROMから読み出されたリードデータを格納するデータバッファと、

リードコマンドを発行して前記フラッシュEEPROMをリード動作モードに設定して前記所定ブロックから第1の書換回数データを読み出し、その第1の書換回数デ

ータに従って前記フラッシュEEPROMの書換回数をブロック毎に管理する書換回数管理手段と、

前記ホスト装置からのライト要求にตอบสนองして、前記フラッシュEEPROMをライトアクセスするライトアクセス手段とを具備し、

このライトアクセス手段は、

リードコマンドを発行して前記フラッシュEEPROMをリード動作モードに設定し、前記ライト要求によって指定されたライト対象ブロック内の前記所定ページから第2の書換回数データを読み出し、その第2の書換回数データの値を更新する手段と、

ライト対象ブロック内におけるライトアクセス対象外のページに格納されているデータを前記データバッファ上に読み出し、その読み出したデータのデータバッファ上のアドレスと前記データバッファに格納されているリードデータのアドレスとに基づいて前記ライト対象のブロックに書き込むべき1ブロック分のブロックデータを構成するデータそれぞれのデータバッファ上の格納位置を示すアドレス情報を保持するヘッダテーブルを生成する手段と、

イレースコマンドを発行して前記フラッシュEEPROMを消去動作モードに設定し、前記ライト対象ブロックの記憶内容を消去する手段と、

前記ヘッダテーブルのアドレス情報を参照して、前記データバッファからページ単位で前記ブロックデータを読み出して前記フラッシュEEPROMのデータレジスタに転送する手段とを含み、

前記書換回数管理手段は、前記第2の書換回数データの更新によってその第2の書換回数データから前記第1の書換回数データへの桁上がりが発生した際、前記ライト対象ブロックに対応する第1の書換回数データの値を更新する手段とを含むことを特徴とする半導体ディスク装置。

【請求項11】 前記書換回数管理手段は、

前記ライト対象ブロックの第1の書換回数データの値が更新された際、そのライト対象ブロックとの書換回数の差が所定値以上に達するブロックをスワッピング先ブロックとして決定する手段と、

前記ライト対象ブロックと前記スワッピング先ブロックのブロックアドレスおよび記憶内容を互いに入れ替えるスワッピング手段とをさらに具備することを特徴とする請求項10記載の半導体ディスク装置。

【請求項12】 前記複数のブロックの所定ページの冗長領域には、前記第2の書換回数データの初期値として、それぞれ離散的な値が格納されていることを特徴とする請求項11記載の半導体ディスク装置。

【請求項13】 前記所定ブロックには、前記複数のブロックの中で最小の第1の書換回数データの値を示す最小書換回数情報と、前記第1の書換回数データの値に応じてスワッピング先ブロックの優先順位を示す優先順位情

報とを保持するスワッピングテーブルが格納されており、

前記書換回数管理手段は、

前記ライト対象ブロックの第1の書換回数データの値が更新された際、そのライト対象ブロックの書換回数と前記スワッピングテーブルの最小書換回数情報の示す値とを比較し、その書換回数の差が所定値以上か否かを判定する手段と、

書換回数の差が所定値以上の時、前記スワッピングテーブルの前記最優先順位情報を参照して最優先度のスワッピング先ブロックを決定し、前記ライト対象ブロックと前記最優先度のスワッピング先ブロックとのブロックアドレスおよび記憶内容を互に入れ替えるスワッピング手段とをさらに具備することを特徴とする請求項10記載の半導体ディスク装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は半導体ディスク装置に関し、特にデータ記憶領域および冗長領域を各々が含む複数のページ領域を有するフラッシュEEPROMを内蔵し、ホスト装置からのディスクアクセス要求に応じて前記フラッシュEEPROMをアクセスする半導体ディスク装置に関する。

【0002】

【従来の技術】従来のワークステーションやパーソナルコンピュータ等の情報処理装置の多くは、2次記憶装置として磁気ディスク装置を用いていた。磁気ディスク装置は、記録の信頼性が高い、ビット単価が安いなどの利点がある反面、装置のサイズが大きい、物理的な衝撃に弱いなどの欠点を持つ。

【0003】そこで、近年、装置のサイズが小さく物理的な衝撃にも強い半導体ディスク装置に注目が集まっている。半導体ディスク装置とは、電気的に一括消去が可能な不揮発性の半導体メモリであるフラッシュEEPROMを、従来の磁気ディスク装置などと同様にパーソナルコンピュータなどの2次記憶装置として用いるものである。この半導体ディスク装置には、磁気ディスク装置の磁気ヘッドや回転ディスクのような機械的な可動部分を含まないため、物理的な衝撃による誤動作や故障が発生しにくい、また、装置としてのサイズも小さくなる等の利点がある。

【0004】ところで、通常、半導体ディスク装置においては、ECC演算機能が設けられており、フラッシュEEPROMに書き込まれるデータには、そのデータ内容に対応するECCが生成されて付加される。このECC演算機能を利用することにより、フラッシュEEPROMに書き込まれるデータの信頼性を高める事ができる。

【0005】ECC演算機能を持つ従来の半導体ディスク装置で採用されている典型的なデータ格納形式を

図21に示す

従来では、ホスト装置によるディスクアクセス単位であるセクタ毎にECCが生成される。このため、図21に示されているように、512バイトのサイズを持つセクタデータがフラッシュEEPROMの2ページに亘って格納される場合には、そのセクタデータに後続する位置にECCが格納されている。セクタデータがフラッシュEEPROMの何ページに亘って格納されるかはフラッシュEEPROMの物理的サイズによって決定されるが、現在開発されている最もサイズの大きい16MビットのフラッシュEEPROMを使用した場合には、512バイトのセクタデータは、図示のように、フラッシュEEPROMの2ページに亘って格納されることになる。

【0006】この場合、その2ページに亘るセクタデータが演算されてECCが生成される。この生成されたECCは、セクタデータに後続して第2ページ目に格納される。

【0007】しかしながら、このようなデータ格納形式を使用した従来の半導体ディスク装置においては、フラッシュEEPROMがもつ以下のようなエラー発生特性を考慮すると、データ記憶の信頼性が低下される等の不具合が生じる。

【0008】すなわち、フラッシュEEPROMを初めとする半導体メモリにおいては、ある特定のメモリセルだけでなく、同一ビット線に接続される複数のメモリセルに同時に不良が発生するというエラーモードが存在する。このエラーモードは、例えば、ビット線の不良や、ビット線とセルとの間のコンタクト不良などによって引き起こされる。

【0009】このようなエラーモードが発生した場合、図21に示されているように、フラッシュEEPROMの複数ページそれぞれの同一ビット位置に不良セルが発生される。この場合、従来の半導体ディスク装置においては、ECC演算の対象となるデータ列に図示のように2箇所エラーが発生されることになる。

【0010】一般に、ECCを利用したエラー検出および訂正では、エラーが1箇所の場合はそのエラーの検出および訂正を容易に行うことができるが、2箇所以上のエラーについてはその検出および訂正が困難となる。このようなエラーに対応するためには、データ回復能力の高い複雑なECC演算式が必要とされる。

【0011】しかし、ECC演算式を複雑にすると、半導体ディスク装置の構成が複雑化されると共に、そのデータライト時のECCの生成およびデータリード時のECCチェック演算に多くの時間を要するため半導体ディスク装置のアクセス速度が低下されるという欠点が引き起こされる。

【0012】また、従来の半導体ディスク装置では、ECCの生成、ECCチェック演算だけでなく、その演算

結果に基づくエラー検出およびエラー訂正処理も全てハードウェアによって実行されていた。このため、そのハードウェアを実現するためのゲート数が増大したり、ハードウェアの複雑化によって結果的にリード/ライト速度が低下されるといった問題があった。

【0013】さらに、従来の半導体ディスク装置では、フラッシュEEPROMの各ブロックの書き換え回数、そのフラッシュEEPROMの特定のブロックに格納された書換回数情報によって一括管理されている。このため、フラッシュEEPROMにデータを書き込む際には、データ書き込みのためのライトアクセスと書換回数情報を更新するためのライトアクセスとの2回のライトアクセスが必要とされる。したがって、データ書き込みに要する時間が長くなると共に、書換回数情報を記憶しているブロックに対するライトアクセスの回数が他のブロックに比べて突出して多くなり、これによってフラッシュEEPROMの寿命が短くなる問題があった。

【0014】

【発明が解決しようとする課題】従来の半導体ディスク装置では、フラッシュEEPROMの物理的構造が十分に考慮されていなかった。このため、ECC演算機能を設けても、十分な記憶信頼性を得る事ができなかった。また、書き換え回数情報が特定のブロックに集中して格納されているので、そのブロックの書き換え回数だけが増大し、これによってフラッシュEEPROMの寿命が短くなる欠点があった。

【0015】この発明はこのような点に鑑みてなされたもので、フラッシュEEPROMの各ページの冗長領域を有効利用してECCや書換回数情報を管理できるようにし、十分な記憶データの信頼性およびフラッシュEEPROMの高寿命化を実現することができる半導体ディスク装置を提供することを目的とする。

【0016】

【課題を解決するための手段および作用】請求項1に係る発明は、データ記憶領域および冗長領域を各々が含む複数のページを有するメモリアルレイと、1ページ分のデータを保持するデータレジスタとを有し、データレジスタとメモリアルレイ間のデータ転送がページ単位で実行されるフラッシュEEPROMを内蔵し、ホスト装置からのディスクアクセス要求に応じて前記フラッシュEEPROMをアクセスする半導体ディスク装置において、前記ホスト装置から転送されるライトデータおよび前記フラッシュEEPROMから読み出されたリードデータを格納するデータバッファと、前記ホスト装置からのライト要求に応じて、前記フラッシュEEPROMをライトアクセスするライトアクセス手段とを具備し、このライトアクセス手段は、前記データバッファに格納されているライトデータを前記各ページのデータ記憶領域のサイズに相当するデータ列単位で演算して、データ列毎に誤り訂正符号を生成する誤り訂正符号生成手

段と、前記データ列およびそれに対応する誤り訂正符号が同一ページのデータ記憶領域および冗長領域に書き込まれるように、データ列毎にそれに対応する誤り訂正符号を付加して前記フラッシュEEPROMのデータレジスタに転送する手段とを含むことを特徴とする。

【0017】この半導体ディスク装置においては、フラッシュEEPROMのページ毎に誤り訂正符号が生成され、その誤り訂正符号が各ページの冗長領域に書き込まれる。このため、たとえフラッシュEEPROMの複数ページそれぞれの同一ビット位置に不良セルが発生しても、誤り訂正符号の演算対象となるデータ列に含まれる不良セルは1つだけとなる。したがって、データ回復能力の高い複雑なECC演算式を利用することなく通常の簡単なECC演算によってエラー検出および訂正を行うことが可能となり、記憶信頼性が高く、且つ高速アクセスが可能な半導体ディスク装置が実現される。

【0018】また、請求項3および請求項4に係る発明では、それぞれ誤り訂正符号だけでなく、ホスト装置から指定される不良セクタ情報やエラー情報も冗長領域に書き込む事を特徴とする。

【0019】これによって、不良セクタ情報やエラー情報を集中管理する必要がなくなるので、それら情報を格納するための専用領域を設ける必要がなくなり、フラッシュEEPROMのデータ記憶領域をユーザデータの格納のために有効利用する事ができる。

【0020】また、請求項5に係る発明は、請求項1の構成に加え、前記ホスト装置によって実行されるプログラムが格納されたROMを具備しており、前記ホスト装置からの所定番地のリード要求にตอบสนองして前記ROMから前記ホスト装置にプログラムが転送されるように構成されていることを特徴とする。

【0021】この半導体ディスク装置においては、フラッシュEEPROMの他に、アプリケーションプログラムやオペレーティングシステムなどのプログラムが格納されたROMが設けられており、このROMのプログラムは、ホスト装置から所定番地のリード要求が発行された時に読み出される。アプリケーションプログラムなどのプログラムは書換が発生しないので、そのプログラムを比較的安価なROMに格納しておく事により、高価なフラッシュEEPROMのデータ記憶領域をユーザデータの格納に有効利用する事ができる。

【0022】また、請求項6に係る発明は、データ記憶領域および冗長領域を各々が含む複数のページを有するメモリアルレイと、1ページ分のデータを保持するデータレジスタとを有し、データレジスタとメモリアルレイ間のデータ転送がページ単位で実行されるフラッシュEEPROMを内蔵し、ホスト装置からのディスクアクセス要求に応じて前記フラッシュEEPROMをアクセスする半導体ディスク装置において、マイクロプロセッサと、前記ホスト装置から転送されるライトデータお

および前記フラッシュEEPROMから読み出されたリードデータを格納するデータバッファと、前記マイクロプロセッサによって制御され、前記フラッシュEEPROMをリード/ライトアクセスするフラッシュメモリインタフェース回路と、前記マイクロプロセッサによってリード/ライト可能な1/0レジスタ群とを具備し、前記フラッシュメモリインタフェース回路は、前記マイクロプロセッサによって前記データバッファから前記各ページのデータ記憶領域のサイズに相当するデータ列単位で読み出されるライトデータを演算して、データ列毎に誤り訂正符号を生成する誤り訂正符号生成回路と、前記データ列およびそれに対応する誤り訂正符号が同一ページのデータ記憶領域および冗長領域に書き込まれるように、データ列毎にそれに対応する誤り訂正符号を付加して前記フラッシュEEPROMのデータレジスタに転送するライトデータ転送回路と、前記フラッシュEEPROMのデータレジスタから読み出されるリードデータから前記データ記憶領域に格納されていたデータ列を取り出して前記データバッファに転送するリードデータ転送回路と、前記リードデータに含まれる誤り訂正符号を利用して前記リードデータの誤り検出を実行し、その検出結果を前記1/0レジスタにセットする誤り訂正符号チェック回路とを含み、前記マイクロプロセッサは、前記1/0レジスタにセットされた演算結果に従って、前記データバッファに転送されたリードデータの誤り訂正を実行することを特徴とする。

【0023】この半導体ディスク装置においては、フラッシュメモリインタフェース回路のハードウェアロジックによって誤り訂正符号の生成および誤り検出のためのチェックが行われ、また、マイクロプロセッサによって実行されるファームウェアによって、リードデータの誤り訂正が行われる。したがって、ファームウェアとハードウェアとによって効率的に負荷分散がなされ、ECCの生成、エラー検出のためのECCチェック演算、エラー訂正を全てハードウェアによって実行する場合よりもハードウェアの構成を単純化できる。

【0024】また、請求項7に係る発明は、データ記憶領域および冗長領域を各々が含む複数のページをそれぞれ有する複数のブロックを含み、各ブロックの所定ページの冗長領域にそのブロックの書換回数を示す書換回数情報が格納されるメモリセルアレイと、1ページ分のデータを保持するデータレジスタとを有し、データレジスタとメモリセルアレイ間のデータ転送がページ単位で実行され、消去動作がブロック単位で実行されるフラッシュEEPROMを内蔵し、ホスト装置からのディスクアクセス要求に応じて前記フラッシュEEPROMをアクセスする半導体ディスク装置であって、前記ホスト装置から転送されるライトデータおよび前記フラッシュEEPROMから読み出されたリードデータを格納するデータバッファと、前記ホスト装置からのライト要求に回答

して、前記フラッシュEEPROMをライトアクセスするライトアクセス手段とを具備し、このライトアクセス手段は、前記ライト要求によって指定されるライト対象ブロック内の前記所定ページから書換回数情報を読み出し、その書換回数情報の値を更新する手段と、前記所定ページに書き込むべきライトデータに前記更新された書換回数情報を付加してページデータを生成し、前記ライトデータおよび前記更新された書換回数情報が前記所定ページのデータ記憶領域および冗長領域に格納されるように、前記ページデータを前記フラッシュEEPROMのデータレジスタに転送する手段とを具備することを特徴とする。

【0025】この半導体ディスク装置においては、各ブロックの所定ページの冗長領域にそのブロックに対応する書換回数情報が格納されており、ライト対象ブロックに対するデータ書込みの際に、そのライト対象ブロックの書換回数情報の値が更新される。その更新された書換回数情報は、所定ページに書き込むべきライトデータに付加されてフラッシュEEPROMのデータレジスタに転送される。このように、各ブロック内で書換回数情報が管理されているので、書換回数情報を特定の領域で集中管理する必要がなくなる。このため、書換回数情報を更新するためのライトアクセスによって特定領域の書換回数が突出して増大するという問題の発生を防止できる。また、同一ブロックに対するライトアクセスだけで、データ書込みと書換回数情報の更新が行う事ができ、結果的にライトアクセス性能の向上を図る事ができる。

【0026】また、請求項8に係る発明は、データ記憶領域および冗長領域を各々が含む複数のページをそれぞれ有する複数のブロックを含み、各ブロックの所定ページの冗長領域にそのブロックの書換回数を示す書換回数情報が格納されるメモリセルアレイと、1ページ分のデータを保持するデータレジスタとを有し、外部からのコマンドにตอบสนองして、ブロック単位で記憶内容を消去する消去動作、メモリセルアレイにページ単位でデータを書込むライト動作、およびメモリセルアレイからページ単位でデータを読み出すリード動作を実行するフラッシュEEPROMを内蔵し、ホスト装置からのディスクアクセス要求に応じて前記フラッシュEEPROMをアクセスする半導体ディスク装置において、前記ホスト装置から転送されるライトデータおよび前記フラッシュEEPROMから読み出されたリードデータを格納するデータバッファと、前記ホスト装置からのライト要求にตอบสนองして、前記フラッシュEEPROMをライトアクセスするライトアクセス手段とを具備し、このライトアクセス手段は、リードコマンドを発行して前記フラッシュEEPROMをリード動作モードに設定し、前記ライト要求によって指定されたライト対象ブロック内の前記所定ページから書換回数情報を読み出し、その書換回数情報の値

を更新する手段と、ライト対象ブロック内にライトアクセス対象外のページが存在する際、そのライトアクセス対象外のページに格納されているデータを前記データバッファ上に読み出し、その読み出したデータのデータバッファ上のアドレスと前記データバッファに格納されているライトデータのアドレスとに基づいて前記ライト対象のブロックに書き込むべき1ブロック分のブロックデータを構成するデータそれぞれのデータバッファの格納位置を示すアドレス情報を保持するヘッダテーブルを生成する手段と、イレースコマンドを発行して前記フラッシュEEPROMを消去動作モードに設定し、前記ライト対象ブロックの記憶内容を消去する手段と、前記ヘッダテーブルのアドレス情報を参照して、前記データバッファから前記所定ページに書き込むべきライトデータを読み出し、そのライトデータに前記更新された書換回数情報を付加してページデータを生成し、前記ライトデータおよび前記更新された書換回数情報が前記所定ページのデータ記憶領域および冗長領域に格納されるように、前記ページデータを前記フラッシュEEPROMのデータレジスタに転送する手段と、前記ヘッダテーブルのアドレス情報を参照して、前記データバッファから前記所定ページ以外に書き込むべきライトデータを順次読み出し、それを前記フラッシュEEPROMのデータレジスタに転送する手段とを含むことを特徴とする。

【0027】この半導体ディスク装置においては、ライト対象ブロック内でライトアクセス対象外のページが存在する場合には、ライトアクセス対象外のページに格納されているいわゆる巻き添え消去データが読み出され、その巻き添え消去データとホストからのライトデータそれぞれのアドレスに基づいてヘッダテーブルが生成される。ヘッダテーブルが生成されると、ライト対象ブロックの記憶内容が消去された後、ヘッダテーブルのアドレス情報に従ってブロックデータがデータバッファからページ単位でデータレジスタに転送され、フラッシュEEPROM内で書込み動作が実行される。このデータ転送において、所定ページに書き込むべきデータについては書換回数情報が付加されて転送される。このため、請求項7の半導体ディスク装置と同様に、同一ブロックに対するライトアクセスだけで、データ書込みと書換回数情報の更新が行う事ができ、結果的にライトアクセス性能の向上を図る事ができる。また、1ブロック分のブロックデータの格納位置を示すアドレス情報を保持するヘッダテーブルが生成されるので、このヘッダテーブルを参照する事により、巻き添え消去データとライトデータとを区別すること無く、データバッファからフラッシュEEPROMへのデータ転送処理を行う事ができる。

【0028】また、請求項9の発明は、請求項8の構成に加え、請求項1の誤り訂正符号生成および演算機能を備えた構成であり、各ブロックの所定ページの冗長領域には、書換回数情報と誤り訂正符号が格納され、他の各

ページの冗長領域には誤り訂正符号が格納される。この半導体ディスク装置においては、冗長領域をより有効に利用する事が可能となる。

【0029】また、請求項10の発明は、請求項7の構成において、各ブロックの書換回数情報をその上位ビット部と下位ビット部に別けて管理し、上位ビット部から構成される第1の書換回数データを所定のブロック内に集中して格納し、下位ビット部から構成される第2の書換回数データをブロック毎に所定ページの冗長領域に分散して格納する構成である。

【0030】この半導体ディスク装置においては、冗長領域に格納されるのは書換回数情報の下位ビットデータだけであるので、冗長領域の使用領域を低減できる。また、下位ビットデータは書換が発生する度に更新する必要があるが、この更新は、書換対象ブロックに対するデータ書込みのためのライトアクセスと同時に実行されるので、請求項7の半導体ディスク装置と同様に、データ書込みのためのライトアクセスと書換回数情報の更新のためのライトアクセスとを別個に行う必要もない。また、書換回数情報の上位ビット部のデータは所定ブロックに集中して格納されているので、全てのブロックから順に書換回数情報を読み出すことなく、1つのブロックからのデータ読み出しだけで全てのブロックの書換回数を管理する事ができる。

【0031】また、請求項11の発明は、請求項10の書換回数管理手段に書換回数の平準化を行うためのスワッピング手段を設けたものである。この半導体ディスク装置においては、書換回数情報の下位ビット部のデータに桁上りが発生した際に、平準化のためのブロック入れ替えを行うか否かが検討される。ライト対象ブロックとの書換回数の差が所定値以上に達するブロックが存在する場合には、そのブロックがスワッピング先ブロックとして決定され、そのブロックとライト対象ブロックとの間でアドレスおよび記憶内容が互いに入れ替えられる。このように、書換回数情報の下位ビット部データの桁上りを条件に平準化のためのブロック入れ替えを行うか否かが検討されるので、各ブロックの書換回数を常時監視する必要がなくなり、書換回数の管理が簡単になる。

【0032】また、請求項12の発明は、ディスク製造時や出荷時などにおける第2の書換回数データの初期値として、ブロック毎にそれぞれ離散的な値を格納したものである。これによって、桁上りが発生する時期がブロック毎にずれるので、複数のブロックに対する書換が連続して発生しても、桁上りに起因するスワッピングの検討処理およびスワッピング処理が集中して発生することを防止する事ができる。

【0033】また、請求項13の発明においては、フラッシュEEPROMの所定ブロックには、第1の書換回数データに加えて、スワッピングテーブルが格納されてい

る。このスワッピングテーブルには、第1の書換回数データの値に応じてスワッピング先ブロックの優先順位を示す優先順位情報が保持されている。したがって、第1の書換回数データをソーティングすること無く、スワッピング先ブロックを容易に特定する事が可能となる。

【0034】

【実施例】以下 図面を参照してこの発明の実施例を説明する。

【0035】図1には、この発明の実施例に係わる半導体ディスク装置の構成が示されている。この半導体ディスク装置10は、ハードディスク装置の代替としてパーソナルコンピュータなどのホストシステムに接続されて使用されるものであり、ホストシステム（ファイルシステム、磁気ディスク装置用のドライバ）からのディスクアクセス要求をフラッシュEEPROMに対するアクセス要求にエミュレートして、半導体ディスク装置10内蔵のフラッシュEEPROMをアクセスする。この半導体ディスク装置10には、最大16個のフラッシュEEPROMチップを搭載することが可能である。これらフラッシュEEPROMは、それぞれ10ソケットを介して半導体ディスク装置10の回路基盤に着脱自在に実装される。図1においては、16個のNAND型フラッシュEEPROM11-1～11-16が搭載されている状態が示されている。半導体ディスク装置10には、さらに、コントローラ12、およびRAM13が設けられている。

【0036】フラッシュEEPROM11-1～11-16は、この半導体ディスク装置10の記録媒体として使用されるものであり、ハードディスク装置の磁気記録媒体に相当する。これらフラッシュEEPROM11-1～11-16はNANDメモリバス14を介してコントローラ12に接続されている。NANDメモリバス14には、8ビット幅のデータバス、および各フラッシュEEPROMの制御信号端子に接続される各種制御信号線が定義されている。I/Oデータバスおよび各種制御信号線は、それぞれフラッシュEEPROM11-1～11-16のデータ入出力端子I/O、および制御信号入力端子CONT（コマンドラッチイネーブル端子CLE）、アドレスラッチイネーブル端子ALE、ライトイネーブル端子WE、リードイネーブル端子REなどを含むに共通接続されている。

【0037】さらに、フラッシュEEPROM11-1～11-16とコントローラ12の間には、チップ選択信号（CS1～CS8）線、およびReady/Busy信号線（R/B1～R/B16）がそれぞれチップ毎に独立して配設されている。

【0038】これらフラッシュEEPROM11-1～11-16はそれぞれ16MビットのNAND型EEPROMであり、図2に示されているように、メモリアルレイ111とデータレジスタ112を備えている。メモ

リセルアレイ111は、8K行×264列×8ビットのビット構成を有し、512個のブロックを有している。データ消去はこのブロック単位で実行することができる。各ブロックは16ページ（行）から構成されており、各ページは、256バイトのデータ記憶領域と8バイトの冗長領域を備えている。データの書込みと読み出しは、25618バイトのデータレジスタ112を介してページ単位で実行される。この半導体ディスク装置においては、各ページの冗長領域はECCや書換回数データの格納に利用される。

【0039】フラッシュEEPROM11-1～11-16はそれぞれライトモード、リードモード、消去モードなどの動作モードを有しており、これら動作モードはコントローラ12からのコマンドによって指定される。

【0040】リードモードにおいては、メモリアルアレイ111からデータレジスタ112へのページ単位のデータ転送（ページリード）がフラッシュEEPROM内部で自動実行される。このデータ転送期間中はReady/Busy信号線はBusy状態に設定され、これによってページリード動作中であることがコントローラ12に通知される。データレジスタ112に転送された1ページ分のデータは、入出力端子I/O0～I/O7から8ビット単位でシリアルに読み出される。1ページ分のデータのシリアルリードが終了すると、次のページのページリードが自動的に実行される。

【0041】ライトモードにおいては、データレジスタ112からメモリアルアレイ111へのページ単位のデータ転送（ページライト）がフラッシュEEPROM内部で自動実行される。このデータ転送期間中はReady/Busy信号線はBusy状態に設定され、これによってページライト動作中であることがコントローラ12に通知される。

【0042】消去モードにおいては、コントローラからのコマンドによって指定された任意のブロックの記憶内容が一括消去される。

【0043】半導体ディスク装置10においては、これらフラッシュEEPROM11-1～11-16の制御のための処理の殆どはファームウェアによって制御される。ハードウェアによって実行されるのは、フラッシュEEPROMへのデータ書込みの際のECC生成と、フラッシュEEPROMからのデータ読み出しの際のECCチェックである。

【0044】次に、半導体ディスク装置10のディスクコントローラ12の構成を説明する。

【0045】ディスクコントローラ12は1個のLSIによって実現されており、そのLSIチップには、図1に示されているように、マイクロプロセッサ（MPU）14、プロセッサバスインタフェース15、ROM16、外部バスインタフェース17、バッファメモリコントローラ18、およびNANDバスインタフェース19

が集積形成されている。

【0046】マイクロプロセッサ14は、ROM16に格納されたファームウェアを実行してこの半導体ディスク装置10全体の動作を制御するものであり、パーソナルコンピュータのCPU1から送られるディスクアクセスのためのホストアドレスを、フラッシュEEPROM11-1~11-16をアクセスするためのメモリアドレスに変換するアドレス変換処理、フラッシュEEPROM11-1~11-16をリード/ライトアクセスするためのアクセス制御処理、フラッシュEEPROM11-1~11-16それぞれのデータ書換回数の管理、およびデータ書換回数を平準化するためのスワッピング処理の制御などを行う。

【0047】ROM13には、マイクロプロセッサ14によって実行されるファームウェア、およびパーソナルコンピュータのCPU1によって実行されるオペレーティングシステムやアプリケーションプログラムなどの各種プログラムが格納されている。オペレーティングシステムやアプリケーションプログラムはROM16内でファイルとして管理されており、それらプログラムはCPU1から半導体ディスク装置10に対して発行される読み出し要求に応じてROM16から読み出され、そしてパーソナルコンピュータのメインメモリ3にロードされる。

【0048】ROM16およびフラッシュEEPROM11-1~11-16には、連続するディスクアドレスがマッピングされており、ROM16もディスク装置10の記憶媒体の一部として利用される。このため、ROM16に格納されているオペレーティングシステムやアプリケーションプログラムの読み出しは、CPU1からの読み出し要求が、ある特定の番地を指定している時のみ実行され、その他の番地を指定している時はフラッシュEEPROM11-1~11-16に対するリードアクセスが行われる。

【0049】外部バスインタフェース17は、IDEやPCMCIA仕様に準拠したインタフェースであり、パーソナルコンピュータのシステムバス2に接続される。外部バスインタフェース17には、CPU1との通信のためのI/Oレジスタ群が設けられている。レジスタ群には、セクタナンバレジスタ171、セクタカウントレジスタ172、データレジスタ173、シリンダレジスタ174、ドライブ/ヘッドレジスタ175、コマンドレジスタ176、ステータスレジスタ177などが含まれている。これらレジスタは、マイクロプロセッサ14およびパーソナルコンピュータのCPU1によってそれぞれリード/ライト可能である。

【0050】セクタナンバレジスタ171には、アクセス先頭位置を指定するセクタ番号がCPU1によってライトされる。セクタカウントレジスタ172には、リード/ライト対象のセクタ数がCPU1によってライトさ

れる。データレジスタ173には、CPU1から供給されるライトデータまたは半導体ディスク装置10から読み出されるリードデータが設定される。シリンダレジスタ174には、リード/ライト対象のシリンダ番号がCPU1によってライトされる。ドライブ/ヘッドレジスタ175には、リード/ライト対象のドライブ番号、およびヘッド番号がCPU1によってライトされる。コマンドレジスタ176には、半導体ディスク装置10の動作を指定するリードコマンドやライトコマンド等がCPU1によってライトされる。ステータスレジスタ177には、CPU1に通知すべき半導体ディスク装置10の各種ステータスがセットされる。

【0051】バッファメモリコントローラ18は、RAM13をアクセス制御するためのものであり、システムバス2とRAM13間のデータ転送などを行う。データ転送には、システムバス2とRAM13間のデータ転送の他、RAM13とフラッシュEEPROM11-1~11-16間のデータ転送がある。後者のデータ転送は、マイクロプロセッサ14によるムーブストリング命令の実行によって行われる。

【0052】NANDバスインタフェース19は、マイクロプロセッサ14の制御の下にフラッシュEEPROM11-1~11-16をアクセス制御するハードウェアロジックであり、フラッシュEEPROM11-1~11-16との間のデータ転送、およびフラッシュEEPROMとの間の各種制御信号の授受を行う。このNANDバスインタフェース19には、マイクロプロセッサ14によってリード/ライト可能なI/Oレジスタ群20と、ECC演算回路21が設けられている。ECC演算回路21は、ECC生成のための演算およびECCチェックのための演算を行う。

【0053】I/Oレジスタ群20には、図3に示されているように、NANDチップナンバレジスタ201、NANDデータリード/ライトレジスタ202、NANDチップイネーブル制御レジスタ203、NANDコマンドラッチイネーブル制御レジスタ204、NANDアドレスラッチイネーブル制御レジスタ205、ECCスタート制御レジスタ206、ECCレディー/ビジーステータスレジスタ207、LWCレジスタ208、6個のECCデータレジスタ209~214、および16個のNANDチップレディー/ビジーステータスレジスタ215-1~215-16などが含まれている。

【0054】NANDチップナンバレジスタ201は、どのNAND型フラッシュEEPROMに対して処理を実行するかを選択するためのレジスタであり、アクセス対象のNAND型フラッシュEEPROMのチップ番号を示す値がプロセッサ14によってセットされる。

【0055】NANDデータリード/ライトレジスタ202は、データバスを介してフラッシュEEPROMに対するデータのリード/ライトを行うためのレジスタで

あり、フラッシュEEPROMに対するコマンドの発行、アドレスの発行、ステータスデータのリードなどに使用される。

【0056】NANDチップイネーブル制御レジスタ203は、フラッシュEEPROMのチップイネーブル端子CEを制御するためのレジスタであり、チップセレクト信号の発生/非発生を示すフラグがプロセッサ14によってセットされる。ビット0にフラグ“1”がセットされると、NANDバスインタフェース19は、NANDチップナンバレジスタ201によって選択されたフラッシュEEPROMのチップイネーブル端子CEをアクティブにするためのチップセレクト信号(CS)を発生する。

【0057】NANDコマンドラッチイネーブル制御レジスタ204は、フラッシュEEPROMのコマンドラッチイネーブル端子CLEを制御するためのレジスタであり、コマンドラッチイネーブル端子CLEをアクティブにするか否かを指定するフラグがプロセッサ14によってセットされる。ビット0にフラグ“1”がセットされると、フラッシュEEPROMのコマンドラッチイネーブル端子CLEがNANDバスインタフェース19によってアクティブに設定される。これにより、データバスにコマンドが発行されたことが、フラッシュEEPROMに対して通知される。

【0058】NANDアドレスラッチイネーブル制御レジスタ205は、フラッシュEEPROMのアドレスラッチイネーブル端子ALEを制御するためのレジスタであり、アドレスラッチイネーブル端子ALEをアクティブにするか否かを指定するフラグがプロセッサ14によってセットされる。ビット0にフラグ“1”がセットされると、フラッシュEEPROMのアドレスラッチイネーブル端子ALEがNANDバスインタフェース19によってアクティブに設定される。これにより、データバスにアドレスが発行されたことが、フラッシュEEPROMに対して通知される。

【0059】ECCスタート制御レジスタ206は、ECC演算の実行開始を制御するためのレジスタであり、RAM13とフラッシュEEPROM11-1~11-16との間のデータ転送を開始する前に、ECC演算の実行開始を指定するフラグがプロセッサ14によってセットされる。フラッシュEEPROMからRAM13へのデータ転送を行う時は、ビット0に“1”がセットされる。この場合、ECC演算回路21は、フラッシュEEPROMから読み出されるデータについてECCチェックを行う。RAM13からフラッシュEEPROMへのデータ転送を行う時は、ビット0に“0”がセットされる。この場合、ECC演算回路21は、RAM13から転送されるデータについてECCの生成を行う。

【0060】ECCレディー/ビジーステータスレジスタ207は、ECC演算回路21によるECC演算(E

CC生成またはECCチェック)が実行中か否かを示すステータスレジスタである。NANDバスインタフェース19は、ECC演算実行中であればビット0に“1”のフラグをセットし、ECC演算の実行が終了すると、ビット0に“0”のフラグをセットする。プロセッサ14は、RAM13とフラッシュEEPROM11-1~11-16との間における256バイトのデータ転送終了後にレジスタ207のフラグを参照してECC演算処理の終了の有無を調べ、そして終了を確認すると、次の256バイトのデータ転送処理を開始する。

【0061】LWCレジスタ208には、フラッシュEEPROMの8バイトの冗長領域の先頭の2バイトに格納すべきデータ、例えば、書換回数データの低位ビットであるLWC(Low Write Count)データがセットされる。この場合、RAM13からフラッシュEEPROMへのデータ転送においては、そのデータ転送に先立って、転送先となるフラッシュEEPROMのライト対象ブロックに対応したLWCデータがファームウェアによってセットされる。フラッシュEEPROMからRAM13へのデータ転送においては、フラッシュEEPROMから読み出されるデータ列からLWCデータがNANDバスインタフェース19によって取り出され、それがセットされる。

【0062】ECCデータレジスタ209~214は、6バイトのECCデータを保持するためのレジスタ群である。RAM13からフラッシュEEPROMへのデータ転送時は、256バイトのライトデータに対応する6バイトのECCデータがECC演算回路21によって生成され、その生成されたECCデータがECCデータレジスタ209~214にセットされる。フラッシュEEPROMからRAM13へのデータ転送時は、ECC演算回路21によって実行されるECCチェック演算の結果がレジスタ209~214にセットされる。このECCチェック演算は、エラー検出を行うために実行される。このECCチェック演算においては、256バイトのリードデータに対応する6バイトのECCデータが再度生成され、その6バイトのECCデータとフラッシュEEPROMから実際に読み出された6バイトのECCデータとの比較がバイト単位で行われる。一致すれば“0”、不一致であれば“1”が対応するレジスタのビット0にセットされる。したがって、エラーが発生してなければ、ECCデータレジスタ209~214のビット0にセットされる値は全て“0”となる。

【0063】NANDチップレディー/ビジーステータスレジスタ215-1~215-16は、16個のフラッシュEEPROM11-1~11-16それぞれのレディー/ビジーステータスを保持するレジスタ群であり、各レジスタのビット0には対応するフラッシュEEPROMがレディー状態であるかビジー状態であるかを示すフラグがNANDインタフェース19によってセッ

トされる。レディー状態ならばフラグ“1”が、ビジー状態であればフラグ“0”がセットされる。レディー状態であるかビジー状態であるかは、フラッシュEEPROM11-1~11-16からそれぞれ独立して入力されるレディー/ビジー信号(R/B1~R/B16)によってチップ毎に判定される。

【0064】図1のRAM13はダイナミックRAMから構成されており、その記憶領域の一部はパーソナルコンピュータのCPU1から転送されるライトデータ、およびフラッシュEEPROM11-1~11-16から読み出されるリードデータを保持するデータバッファ131として使用される。また、RAM13は、マイクロプロセッサ14の作業領域として使用され、ここには、ホストアドレスをメモリアドレスに変換するためのアドレス変換テーブル132、フラッシュEEPROMに書き込むべき1ブロック分のブロックデータの格納位置を管理するヘッダテーブル133、フラッシュEEPROM11-1~11-16の書換回数をブロック単位で管理するための書換回数管理テーブル134などが格納される。

【0065】次に、図4を参照して、マイクロプロセッサ14によって管理される半導体ディスク装置10内のメモリマップについて説明する。

【0066】マイクロプロセッサ14が管理するメモリアドレス空間には、図示のように、RAM領域、NANDメモリデータ領域、アプリケーションROM(APL-ROM)領域、およびファームウェアROM(FW-ROM)領域がマッピングされている。

【0067】RAM領域はRAM13をアクセスするためのメモリアドレス空間であり、ここにはデータバッファ131などが割り当てられている。

【0068】NANDメモリデータ領域は、フラッシュEEPROM11-1~11-16をアクセスするための256バイトのアドレス空間である。フラッシュEEPROM11-1~11-16に対するデータのリード・ライトアクセスにおいては、マイクロプロセッサ14のムーブストリング命令によって256バイトのデータ転送が行われる。このため、この256バイトのデータ転送のために、256バイトのアドレス空間がNANDメモリデータ領域として確保されている。このNANDメモリデータ領域を使用する事により、データバッファ131とフラッシュEEPROM11-1~11-16との間のデータ転送は全てファームウェアの処理として行う事ができる。

【0069】次に、図5を参照して、フラッシュEEPROM11-1~11-16それぞれのブロックの利用形態について説明する。

【0070】フラッシュEEPROM11-1~11-16のブロックはユーザデータの格納だけでなく、これらフラッシュEEPROMを制御するための各種管理情

報の格納にも利用されている。フラッシュEEPROM11-1~11-16に対するユーザデータと管理情報の割り当ては以下の通りである。

【0071】すなわち、チップ番号#0のNAND型フラッシュEEPROM11-1においては、図示のように、512個のブロックの中でブロック0~ブロック503の504個のブロックは、ユーザデータを格納するためのメモリブロックとして利用される。また、ブロック504は、チップ内の512個のブロックのアドレスや書換回数などを管理するための管理情報を記憶するMBI(Memory Block Information)テーブルブロックとして利用される。また、ブロック505~509は、メモリブロックおよびMBIテーブルブロックに不良が発生した時にその代替用の予備ブロックとして使用される。ブロック511は半導体ディスク装置10の構成情報を記憶するコンフィグブロックとして使用され、ブロック510はコンフィグブロックの代替用の予備ブロックとして使用される。

【0072】コンフィグブロックは装置10全体で1個だけ存在し、ここには、半導体ディスク装置10に搭載されているフラッシュEEPROMのチップ数や、各フラッシュEEPROMにおけるMBIテーブルブロックの存在位置などを示す情報などが管理されている。

【0073】チップ番号#1~#16のNAND型フラッシュEEPROM11-2~11-16においては、図示のように、ブロック0~ブロック503の504個のブロックは、ユーザデータを格納するためのメモリブロックとして利用され、ブロック504は、チップ内の512個のブロックのアドレスや書換回数を管理するための管理情報を記憶するMBI(Memory Block Information)テーブルブロックとして利用される。また、ブロック505~511は、メモリブロックおよびMBIテーブルブロックの代替用のブロックとして使用される。

【0074】メモリブロック、MBIテーブルブロック、代替ブロック、コンフィグブロックそれぞれの物理的なブロック構造は図2で説明した通りであるが、以下、それらブロック内のデータ格納形式について説明する。

【0075】図6には、ユーザデータを格納するためのメモリブロックのデータ格納形式が示されている。

【0076】図2で説明したように、この半導体ディスク装置10で使用されるNAND型フラッシュEEPROMは16Mビットのチップであり、各ブロックは、ページ0からページ15までの16ページから構成されている。各ページは、256バイトのデータ領域と8バイトの冗長領域とから構成されている。

【0077】一般に、ホストシステムによるディスクアクセスは512バイトのセクタ単位で行われる。この場合、各ページのデータ領域に格納される256バイトの

ユーザデータのサイズは、1セクタ分のユーザデータサイズの半分となる。この半導体ディスク装置10においては、ユーザデータの格納には256バイトのデータ領域だけが使用され、8バイトの冗長領域はユーザデータの格納には利用されない。したがって、この半導体ディスク装置10においては、512バイトのユーザデータは、ブロック内の連続する2ページに互って格納される。このため、1ブロックに格納されるユーザデータは、8セクタ分のデータ、例えば、セクタ0からセクタ7までのデータとなる。

【0078】すなわち、図5に示されているように、ページ0にはセクタ0の前半の256バイト、ページ1にはセクタ0の後半の256バイト、ページ2にはセクタ1の前半の256バイト、ページ3にはセクタ1の後半の256バイトが格納され、以下同様にして、1/2セクタ単位でユーザデータが格納される。

【0079】ページ0～15それぞれの8バイトの冗長領域は、2バイトの領域と6バイトの領域に分割されており、その6バイトの領域にはECCが格納される。このECCは、同一ページのデータ領域に格納されている256バイトのユーザデータに対応するものである。

【0080】すなわち、ページ0の冗長領域に格納されているECCは、ページ0のデータ領域に格納されている256バイトのユーザデータから生成されたものであり、同様に、ページ1の冗長領域に格納されているECCは、ページ1のデータ領域に格納されている256バイトのユーザデータから生成されたものである。

【0081】このように、この半導体ディスク装置10においては、512バイトのセクタ単位ではなく256バイトのページ単位でECCが生成され、演算対象のデータ列とECCとが同一ページに格納される。このため、たとえ、図示のようにフラッシュEEPROMの複数ページそれぞれの同一ビット位置に不良セルが発生するというエラーが発生しても、ECCの演算対象となるデータ列に含まれる不良セルは1つだけとなる。したがって、データ回復能力の高い複雑なECC演算式を利用することなく通常の簡単なECC演算によってエラー検出および訂正を行うことが可能となる。

【0082】また、ページ0の冗長領域の残りの2バイトには、ブロック0のブロックステータス情報として、LWCデータが格納されている。このLWCデータは、ブロック0の書換回数を示す書換回数データの低位ビット部、例えば26ビット幅の書換回数データの低位10ビットのデータから構成される。

【0083】LWCデータはブロック毎に1つ格納され、ブロック1～ブロック503それぞれの先頭ページの冗長領域の先頭の2バイトにも、同様にして、そのブロックステータス情報としてLWCデータが格納されている。

【0084】さらに、ページ1の冗長領域の残りの2バ

イトには、セクタ0のセクタステータス情報として、BADセクタ情報、およびECCエラー情報が格納されている。

【0085】BADセクタ情報は、そのセクタ0が不良セクタか否かを示すステータス情報である。同様に、ページ3、ページ5、……ページ15それぞれの冗長領域の残りの2バイトにも、セクタ1、セクタ2、……セクタ7それぞれに対応するBADセクタ情報が格納されている。これらBADセクタ情報は、半導体ディスク装置10のホストシステムであるパーソナルコンピュータが、半導体ディスク装置10をフォーマットするためのフォーマットコマンドを実行する時に、そのパーソナルコンピュータのCPUによって与えられる。

【0086】ECCエラー情報は、そのセクタに対するECCエラー発生の有無を示すステータス情報である。このECCエラー情報は、例えば、パーソナルコンピュータがWRITE LONGコマンドによるデータ書き込みを行った場合においてECCエラーとなるパーストエラーなどのエラーが発生した時に、パーソナルコンピュータのCPUによって与えられる。

【0087】次に、図7を参照して、MBIテーブルブロックのデータ格納形式を説明する。

【0088】MBIテーブルブロックは、ページ0からページ15までの16ページを含む1個のブロックから構成されている。各ページは、256バイトのデータ領域と8バイトの冗長領域とから構成されている。8バイトの冗長領域のうちの後の6バイトには、そのページのデータ領域に格納されている256バイトのデータに対応するECCが格納されている。また、冗長領域の先頭の2バイトは未使用である。

【0089】MBIテーブルブロックのデータ領域には、図示のように、アドレス変換テーブル、UWCテーブル、およびスワップオーダーテーブルが格納されている。

【0090】MBIテーブルブロックにおけるアドレス変換テーブルは、チップ内のブロック0～511の物理ブロックアドレス0～511を、シリンダ番号、ヘッド番号およびセクタ番号によって規定されるホストアドレスに変換するためのアドレス変換情報を保持する。ホストアドレスは、物理ブロックアドレス順に格納されている。

【0091】ここで、MBIテーブルブロックにおけるホストアドレスの値としては、ホストシステムから与えられるシリンダ番号、ヘッド番号およびセクタ番号そのものではなく、それらシリンダ番号、ヘッド番号およびセクタ番号から計算されたホスト論理アドレスが使用される。

【0092】このホスト論理アドレスは、フラッシュEEPROM11-1～11-16に設けられる全セクタ(=8×512×16セクタ)に対して“0”から順に

10

20

30

40

50

連続するセクタ番号を割り当てた場合のセクタの通し番号である。したがって、ホスト論理アドレスのビット幅は16ビットとなる。しかし、このMB1テーブルブロックのアドレス変換テーブルで管理されるアドレス変換情報は、1セクタ単位ではなく、1ブロックに含まれるセクタ数である8セクタ単位でアドレス変換を行うためのものであるため、実際には、16ビットのホスト論理アドレスのなかの上位13ビットだけがテーブル上に格納されている。

【0093】UWCテーブルは、チップ内の512個のブロックそれぞれの書換回数を示す書換回数データの上位ビット部から構成されるUWC (UPPER WRITING COUNT) データを保持する。UWCデータは16ビット幅を有しており、このUWCデータと前述の13ビット幅のLWCとを合わせると、最大で6400万回数の書換までカウントする事ができる。UWCデータの値は、対応するLWCデータに桁上がりが発生する度、つまり1024回の書換が発生する度に、+1カウントアップされる。

【0094】これらアドレス変換テーブルおよびUWCテーブルは、図示のように、MB1テーブルブロックのページ5からページ12のデータ領域に格納されている。アドレス変換テーブルおよびUWCテーブルは、装置起動時、つまり半導体ディスク装置10のパワーオン時に、RAM13に転送されて、それぞれアドレス変換テーブル132および書換回数管理テーブル134として常駐される。

【0095】スワップオーダーテーブルは、ブロック入れ替えのためのスワッピング操作の際にマイクロプロセッサ14によって参照されるテーブルであり、スワッピング先のブロック(被スワップブロック)を示すブロックアドレスを、優先度順(UWCデータの値が小さい順)に保持している。スワップオーダーテーブルは、図示のように、MB1テーブルブロックのページ13からページ15のデータ領域に格納されている。スワップオーダーテーブルは、RAM13には常駐されず、ブロック入れ替えの検討時、つまりLWCデータからUWCデータへの桁上がり発生時に、RAM13に転送されてマイクロプロセッサ14によって参照される。

【0096】図8には、アドレス変換テーブルとUWC

データの格納領域はそれぞれ2バイトのデータサイズを有している。ここで、ホストアドレスは前述したように13ビットであるので、2バイトの格納領域の上位3ビット(ビット15～ビット13)は使用されない。ホストアドレスとUWCデータは、物理ブロックアドレス順に交互に配置されている。

【0098】すなわち、MB1テーブルブロックにおけるページ5の256バイトのデータ領域には、その先頭

から順に、物理ブロックアドレス0に対応するホストアドレス、物理ブロックアドレス0に対応するUWCデータ、物理ブロックアドレス1に対応するホストアドレス、物理ブロックアドレス1に対応するUWCデータといった順番で、ホストアドレスとUWCデータが64個のブロック(ブロック0～ブロック63)分格納されている。

【0099】同様に、ページ12の256バイトのデータ領域には、その先頭から順に、物理ブロックアドレス448に対応するホストアドレス、物理ブロックアドレス448に対応するUWCデータ、物理ブロックアドレス449に対応するホストアドレス、物理ブロックアドレス449に対応するUWCデータといった順番で、ホストアドレスとUWCデータが64個のブロック(ブロック448～ブロック511)分格納されている。

【0100】図9には、スワップオーダーテーブルの構成が示されている。

【0101】スワップオーダーテーブルは、ブロック0からブロック255までの前半の256個のブロックに対応する第1オーダーテーブルと、ブロック256からブロック511までの後半の256個のブロックに対応する第2オーダーテーブルとに分割されている。

【0102】第1オーダーテーブルにおいて、その先頭の2バイトの領域(アドレス800h, 801h)には、ブロック0からブロック255の中の最小のUWCの値が格納されている。

【0103】次の1バイト(アドレス802h)には、最小UWCの値を持つブロックの総数を示す値が格納される。

【0104】さらに、次の1バイト(アドレス803h)には、最小UWC+1の値を持つブロックの総数を示す値が格納される。ここで、“最小UWC+1”とは、あくまでも最小UWCに1を加えた値を示しており、最小UWCの次に小さなUWCを示すものではない。最小UWC+1の値を持つブロック総数がゼロとなる場合もある。

【0105】以降、アドレス804h～87Ahには、最小UWC+2の値を持つブロック総数を示すデータから最小UWC+120の値を持つブロック総数を示すデータが順次格納される。

【0106】アドレス87Ch～87Fhには、MB1テーブルブロック自体の書換回数を示すデータなどが格納される。

【0107】アドレス880h～97Fhには、本来のスワップオーダーテーブルとしての情報が格納されている。すなわち、アドレス880hには、ブロック0～ブロック255の中で最優先度の被スワップ対象のブロックを示す値として、最小UWCの値を有するブロックの物理ブロックアドレスが格納されている。最小UWCの

値を有するブロックが複数存在する場合には、物理ブロックアドレスが最も後ろのブロックから順に優先順位が高くなる。以下、同様にして、アドレス881hから97Fhに、優先度順にスワップ対象ブロックのブロックアドレスが格納される。

【0108】次に、第2オーダーテーブルについて説明する。

【0109】第2オーダーテーブルにおいて、その先頭の256バイトの領域（アドレス98Ch, 981h）には、MB1テーブルブロック、代替ブロック、コンヒギブロックを除き、ブロック256からブロック511の中の最小のUWCの値が格納されている。すなわち、代替ブロック等は書換回数が全く進行していないので、それらブロックの書き替え回数は無視される。また、実際に代替が発生した場合は、その代替ブロックのUWCデータの値は、その代替ブロックの実際の書換回数ではなく、代替元、つまり欠陥ブロックのUWCを譲り受けることとする。これは、スワッピング操作の多発による性能低下を防止するためである。もし、代替ブロックの実際のUWCデータの値をそのまま採用すると、書換回数の値が突出して小さなブロックがいきなり登場することになり、その代替ブロックをスワッピング被対象とするスワッピング操作が何度も発生されてしまうことになる。

【0110】次の1バイト（アドレス982h）には、最小UWCの値を持つブロックの総数を示す値が格納される。

【0111】さらに、次の1バイト（アドレス983h）には、最小UWC+1の値を持つブロックの総数を示す値が格納される。以降、アドレス984h~9FAhには、最小UWC-2の値を持つブロック総数を示すデータから最小UWC+120の値を持つブロック総数を示すデータが順次格納される。

【0112】アドレスA00h~AFFhには、本来のスワップオーダーテーブルとしての情報が格納されている。すなわち、アドレスA0Chには、ブロック256~ブロック511の中で最優先度の被スワップ対象のブロックを示す値として、最小UWCの値を有するブロックの物理ブロックアドレスが格納されている。最小UWCの値を有するブロックが複数存在する場合には、物理ブロックアドレスが最も後ろのブロックから順に優先順位が高くなる。以下、同様にして、アドレスA01hからAFFhに、優先度順にスワップ対象ブロックのブロックアドレスが格納される。

【0113】このように、スワップオーダーテーブルにおいては、スワッピング先となるブロックがその優先度順に格納されている。このため、このスワップオーダーテーブルを参照する事により、マイクロプロセッサ14は、UWCデータのソーティングを行うことなく、スワッピング先を特定する事が可能となる。

【0114】図10には、コンヒギブロックのデータ格納形式が示されている。

【0115】コンヒギブロックは、ページ0からページ15までの16ページを含む1個のブロックから構成されている。各ページは、256バイトのデータ領域と8バイトの冗長領域とから構成されている。8バイトの冗長領域のうちの6バイトには、そのページのデータ領域に格納されている256バイトのデータに対応するECCが格納されている。また、冗長領域の残りの2バイトは未使用である。

【0116】コンヒギブロックのデータ領域には、図示のように、コンヒギ情報、CIS情報、ドライブID情報が格納されている。

【0117】コンヒギ情報は、この半導体ディスク装置10に搭載されているフラッシュEEPROMのチップ数、同時書込み可能な最大チップ数、各チップにおけるMB1テーブルブロックの存在位置を示す物理ブロックアドレスなどを定義する。このコンヒギ情報は半導体ディスク装置10のパワーオン時にマイクロプロセッサ14によって参照され、そのコンヒギ情報に従った初期設定処理が行われる。

【0118】CIS情報は、半導体ディスク装置10をPCMCIA仕様のカードとして使用する場合に、ホストシステムに通知されるカード属性情報である。このカード属性情報には、メモリチップの種類、容量、アクセスタイムなどの物理的仕様に関する情報と、このカードがディスク装置であることなどを示す用途に関する情報が定義されている。このCIS情報は、半導体ディスク装置10のパワーオン時に、マイクロプロセッサ14によってRAM13に転送される。RAM13に転送されたCIS情報は、ホストシステムからの所定番地の読み出し要求に回答してホストシステムに転送される。

【0119】ドライブID情報は、半導体ディスク装置10のドライブ構成を示す情報であり、ここには、内蔵フラッシュEEPROMに論理的に割り当てられる総セクタ数、シリンダ数、ヘッド数、1トラック当たりのセクタ数などが定義されている。このドライブID情報は、ホストシステムからのドライブ識別コマンドに回答して、マイクロプロセッサ14によってホストシステムに転送される。

【0120】このように、この実施例においては、ユーザデータを格納するためのメモリブロックだけでなく、管理情報格納のために使用されるMB1テーブルブロックやコンヒギブロックにおいても、各ページの冗長領域には、そのページのデータ領域に格納されるデータに対応したECCが格納されている。したがって、これらMB1テーブルブロックおよびコンヒギブロックにおいても、データ回復能力の高い複雑なECC演算式を利用することなく通常の簡単なECC演算によってエラー検出および訂正を行うことが可能となる。

【0121】次に、図11を参照して、RAM13に設けられるアドレス変換テーブル132の構成について説明する。

【0122】このアドレス変換テーブル132は、シリンダ番号、ヘッド番号およびセクタ番号によって規定されるホストアドレスを、フラッシュEEPROM11-1~11-116をアクセスするための物理メモリアドレスに変換するためのものであり、このアドレス変換テーブル132はフラッシュEEPROM11-1~11-116それぞれのMB1テーブルブロックのアドレス変換情報に基づいて生成される。

【0123】このアドレス変換テーブル132においては、ホストアドレスとしては13ビットのブロックセクタアドレスが使用される。このブロックセクタアドレスは、前述したように、シリンダ番号、ヘッド番号およびセクタ番号から計算されたセクタの通し番号を示すホスト論理アドレス（16ビット）の中の上位13ビットから構成される。また、物理メモリアドレスは、4ビットのチップ番号アドレスと、9ビットのチップ内物理ブロックアドレスから構成される。4ビットのチップ番号アドレスは、チップ番号#0のフラッシュEEPROM11-1からチップ番号#15のフラッシュEEPROM11-8までの16個のチップのうちの1つを指定する。9ビットのチップ内物理ブロックアドレスは、チップ番号アドレスによって指定されたチップ内のブロック0~ブロック511の512個のブロックの中の1つを指定する。物理メモリアドレス（チップ番号アドレス、チップ内物理ブロックアドレス）は、ホストアドレス順に格納されている。

【0124】次に、図12を参照して、アドレス変換テーブル132を利用したアドレス変換動作の原理を説明する。

【0125】まず、ホストシステムから指定されたシリンダ番号、ヘッド番号およびセクタ番号は、マイクロプロセッサ14によって実行されるファームウェアによって16ビット幅のホスト論理アドレスに変換される。次いで、その16ビット幅のホスト論理アドレスの上位13ビットから成るブロックセクタアドレスをインデックスとしてアドレス変換テーブル132が検索され、ブロックセクタアドレスに対応する4ビット幅のチップ番号アドレス、および9ビット幅の物理ブロックアドレスが決定される。

【0126】チップ番号アドレスは、チップ選択のために、ファームウェアによってNANDバスインタフェース19のNANDチップナンバレジスタ201にセットされる。また、9ビットの物理ブロックアドレスには、16ビット幅のホスト論理アドレスの下位3ビットから成るブロック内セクタアドレスと、1ビット幅の偶数/奇数ページ指定アドレスと、8ビット幅のカラムアドレスがNANDバスインタフェース19内で付加されて、

20ビット幅のメモリアドレス（A20-A0）が生成される。このメモリアドレス（A20-A0）は、アドレス指定のために、NANDバスインタフェース19のNANDデータリード/ライトレジスタ202にセットされる。ユーザデータをリード/ライトするためにホストシステムによって実行される半導体ディスク装置10のアクセスはセクタ単位で行われるので、通常、1ビット幅の偶数/奇数ページ指定アドレス（A8）と、8ビット幅のカラムアドレス（A7-A0）はどちらもオール“0”である。

【0127】次に、図13を参照して、NANDバスインタフェース19のECC演算回路21のECC生成/チェック動作を説明する。

【0128】まず、図13（a）を参照して、ECCの生成動作を説明する。

【0129】NAND型フラッシュEEPROM11-1~11-16に対するデータ書込み時においては、RAM13のデータバッファ131に格納されているユーザデータは、マイクロプロセッサ14によって256バイト単位で読み出されて、NANDバスインタフェース19を介してフラッシュEEPROMのデータレジスタに転送される。

【0130】NANDバスインタフェース19においては、256バイトのユーザデータはECC演算回路21に送られ、そこでECCを生成するための演算が実行される。ここで、ECCとしては例えばハミングコードなどが利用される。

【0131】ECC演算は、256バイトのユーザデータだけでなく、実際には、その256バイトのユーザデータと冗長領域の最初の2バイトに格納すべきデータとから構成される258バイトのデータを対象として行われる。

【0132】例えば、各ブロックの先頭ページに対するデータ書込みの場合には、256バイトのユーザデータと2バイトのLWCデータとから構成される258バイトのデータがECC演算の対象となる。この2バイトのLWCデータは、256バイトのユーザデータの転送に先立って、マイクロプロセッサ14によってLWCレジスタ208にセットされている。

【0133】ECC演算回路21は、256バイトのユーザデータとLWCレジスタ208にセットされている2バイトのLWCデータとから構成される258バイトのデータを所定のデータ列単位で演算して、6バイトのハミングコードECCを順次生成する。これらハミングコードECCはECCデータレジスタ209~214に一時的に保持される。

【0134】ECCの生成処理が完了すると、ECC演算回路21は、256バイトのユーザデータに後続させて、2バイトのLWCデータと6バイトのECCデータを、フラッシュEEPROMのデータレジスタに転送す

る。これにより、フラッシュEEPROMのデータレジスタには、256バイトのユーザデータ、2バイトのLWCデータ、および6バイトのECCデータから構成される1ページ分のデータがセットされる。

【0135】次に、図13(b)を参照して、ECCのチェック動作を説明する。

【0136】NAND型フラッシュEEPROM11〜1〜16からのデータ読み出し時においては、フラッシュEEPROMから読み出される1ページ分のデータがNANDバスインタフェース19に送られる。

【0137】NANDバスインタフェース19においては、1ページ分のデータから256バイトのユーザデータが取り出され、それがRAM13のデータバッファ131に送られる。また、その1ページ分のデータはECC演算回路21に送られる。

【0138】ECC演算回路21は、まず、1ページ分のデータからLWCデータを取り出して、それをLWCレジスタ208にセットし、その後、エラー検出のためのECCチェック演算を実行する。

【0139】このECCチェック演算においては、ECC演算回路21は、256バイトのユーザデータと2バイトのLWCデータとから構成される258バイトのデータをECC生成処理の場合と同じデータ列単位で演算して、6バイトのハミングコードECCを順次生成する。そして、その生成したハミングコードECCと、フラッシュEEPROMから読み出された1ページ分のデータに含まれているハミングコードECCとをバイト単位で比較し、一致／不一致を示す比較結果をECCデータレジスタ209〜215にセットする。エラー発生がない場合には、ECCデータレジスタ209〜215には全て、一致を示すデータ“0”がセットされる。ECCデータレジスタ209〜215の内容はマイクロプロセッサ14によって参照され、エラーが発生していれば、マイクロプロセッサ14によってデータバッファ131上のデータのエラー訂正が行われる。

【0140】次に、マイクロプロセッサ14によって実行されるファームウェアとNANDバスインタフェース19の機能分担を明瞭にするために、フラッシュEEPROMに対するアクセス制御動作を説明する。

【0141】まず、図14および図15を参照して、フラッシュEEPROMのプログラムサイクルの制御に必要なファームウェアの処理とNANDバスインタフェース19の動作について説明する。

【0142】図14はプログラムサイクルにおけるファームウェアの処理とNANDバスインタフェース19の動作を説明するフローチャートであり、図15はプログラムサイクルにおけるフラッシュEEPROMの動作を示すタイミングチャートである。

【0143】ここでは、あるブロックの先頭ページに、256バイトのユーザデータ、2バイトのLWCデ

タ、および6バイトのECCデータをライトする場合を例にとって説明する。

【0144】マイクロプロセッサ14によって実行されるファームウェアは、まず、ライト対象のフラッシュEEPROMチップに対応するNANDチップレディ／ビジーレジスタ215-1〜215-16の1つを参照して、そのライト対象のフラッシュEEPROMチップがレディステート（待機中）か否かを確認する（ステップA11）。レディステートであることを確認すると、ファームウェアは、NANDチップナンバレジスタ201をI/Oライトして、そこにライト対象フラッシュEEPROMチップに対応するチップ番号をセットする（ステップA12）。

【0145】次いで、ファームウェアは、NANDチップイネーブル制御レジスタ203をI/Oライトして、そこにチップイネーブル入力（CE）をアクティブにすることを指定するCEオンコマンドをセットする（ステップA13）。

【0146】この時、NANDバスインタフェース19は、NANDチップナンバレジスタ201にセットされたチップ番号に対応するチップセレクト信号CS1〜CS16の1つをアクティブにする（ステップB11）。このチップセレクト信号はライト対象チップのチップイネーブル入力CE[−]に供給され、これによってライト対象チップがイネーブル状態に設定される。

【0147】次に、ファームウェアは、NANDコマンドラッチイネーブル制御レジスタ204をI/Oライトして、そこにコマンドラッチイネーブル（CLE）信号をアクティブにすることを指定するCLEオンコマンドをセットする（ステップA14）。

【0148】NANDバスインタフェース19は、CLEオンコマンドのセットにตอบสนองして、コマンドラッチイネーブル信号CLEをアクティブ状態、つまり“H”レベルに設定する（ステップB12）。

【0149】次いで、ファームウェアは、NANDデータリード／ライトレジスタ202をI/Oライトして、そこにシリアルデータ入力コマンド“80h”をセットする（ステップA15）。NANDバスインタフェース19は、シリアルデータ入力コマンド“80h”のセットにตอบสนองして、ライトイネーブル信号WE[−]をアクティブ状態、つまり“L”レベルに設定し、且つシリアルデータ入力コマンド“80h”を入出力端子I/Oに転送する（ステップB13）。

【0150】ファームウェアは、NANDアドレスラッチイネーブル制御レジスタ205をI/Oライトして、そこにALEオンコマンドをセットする（ステップA16）。NANDバスインタフェース19は、ALEオンコマンドのセットにตอบสนองして、アドレスラッチイネーブル信号ALEをアクティブ状態、つまり“H”レベルに設定する（ステップB14）。

【0151】次いで、ファームウェアは、NANDデータリード/ライトレジスタ202を1/Oライトして、そこにメモリアドレス(カラムアドレスA7-A0、およびページアドレスA20-A8)をセットする(ステップA17)。NANDバスインタフェース19は、アドレスのセットにตอบสนองして、ライトイネーブル信号WE⁻をアクティブ状態、つまり“L”レベルに設定し、且つセットされたメモリアドレスを入出力端子1/Oに転送する(ステップB15)。

【0152】ファームウェアは、このプログラムサイクル実行前に予めライト対象ブロックから読み出してメモリカウンタアップしておいたLWCデータをLWCレジスタ208にセットする(ステップA18)。そして、ファームウェアは、ECC開始コマンドをECCスタート制御レジスタ206にセットし(ステップA19)、その後、ムーブストリング命令の実行によって、データバッファ131からフラッシュEEPROMに256バイトのユーザデータを転送する(ステップA20)。

【0153】NANDバスインタフェース19は、ECC開始コマンドにตอบสนองしてECC生成のための演算を開始し、256バイトのユーザデータおよびLWCデータに対応する6バイトのECCデータを生成する(ステップB16)。そして、NANDバスインタフェース19は、256バイトのユーザデータに後続して、2バイトのLWCデータおよび6バイトのECCデータの転送を開始する(ステップB17)。これにより、ライト対象フラッシュEEPROMのデータレジスタには、図15に示されているように、256バイトのユーザデータ、2バイトのLWCデータおよび6バイトのECCデータが1ページ分のシリアル入力データ(DIN)として転送される。

【0154】その後、ファームウェアは、CLEオンコマンドを発行した後に、NANDデータリード/ライトレジスタ202を1/Oライトして、そこにプログラムコマンド“10h”をセットする(ステップA21)。NANDバスインタフェース19は、まず、コマンドラッチイネーブル信号CLEをアクティブ状態に設定してからライトイネーブル信号WE⁻をアクティブ状態に設定し、且つプログラムコマンド“10h”を入出力端子1/Oに転送する(ステップB18)。

【0155】このプログラムコマンド“10h”が転送されると、ライト対象のフラッシュEEPROMにおいては、ページライト動作およびベリファイ動作が自動実行される。

【0156】ページライト動作では、データレジスタからメモリアルレイへの1ページ分のページデータ転送が行われて、データレジスタの256バイトのユーザデータがそのページのデータ領域に書き込まれ、且つデータレジスタの2バイトのLWCデータおよび6バイトのECCデータがそのページの冗長領域に書き込まれる。

このページ書き込み動作期間中そのチップからのレディー/ビジー信号(RDY/BSY)はビジー状態を示し、書き込み動作が完了するとビジー状態からレディー状態に戻る。

【0157】ファームウェアは、ライトアクセスしたチップに対応するNANDチップレディー/ビジーレジスタの内容によって書き込み動作の完了を確認すると(ステップA22)、NANDデータリード/ライトレジスタ202を1/Oライトして、そこにステータスリードコマンド“70h”をセットする(ステップA23)。NANDバスインタフェース19は、ステータスリードコマンド“70h”のセットにตอบสนองして、ライトイネーブル信号WE⁻をアクティブ状態、つまり“L”レベルに設定し、且つステータスリードコマンド“70h”を入出力端子1/Oに転送し、その後、リードイネーブル信号RE⁻をアクティブ状態、つまり“L”レベルに設定する(ステップB19)。

【0158】ステータスリードコマンド“70h”が転送されると、フラッシュEEPROMからはページライト動作が成功したか否かを示すステータスデータが読み出され、それがNANDデータリード/ライトレジスタ202にセットされる。

【0159】ファームウェアは、NANDデータリード/ライトレジスタ202のステータスデータを参照してページライト動作が成功したか否かを確認し、成功していればライト動作を終了する(ステップA24)。もし失敗した場合には、代替ブロックとの代替処理などを行う。

【0160】次に、図16および図17を参照して、フラッシュEEPROMのリードサイクルの制御に必要なファームウェアの処理とNANDバスインタフェース19の動作について説明する。

【0161】図16はリードサイクルにおけるファームウェアの処理とNANDバスインタフェース19の動作を説明するフローチャートであり、図17はリードサイクルにおけるフラッシュEEPROMの動作を示すタイミングチャートである。

【0162】フラッシュEEPROM11-1~11-16のリードサイクルには、ページ単位でデータを読み出す第1リードサイクルと、各ページの冗長領域のデータだけを読み出す第2リードサイクルとがある。

【0163】ここでは、第1リードサイクルで、あるブロックの先頭ページを読み出す場合を説明する。

【0164】マイクロプロセッサ14によって実行されるファームウェアは、まず、リード対象のフラッシュEEPROMチップに対応するNANDチップレディー/ビジーレジスタ215-1~215-16の1つを参照して、そのリード対象のフラッシュEEPROMチップがレディーステート(待機中)か否かを確認する(ステップA31)。レディーステートであることを確認する

10

20

30

40

50

と、ファームウェアは、NANDチップナンバレジスタ201を1/Oライトして、そこにリード対象フラッシュEEPROMチップに対応するチップ番号をセットする(ステップA32)。

【0165】次いで、ファームウェアは、NANDチップイネーブル制御レジスタ203を1/Oライトして、そこにチップイネーブル入力(CF)をアクティブにすることを指定するCEオンコマンドをセットする(ステップA33)。

【0166】この時、NANDバスインタフェース19は、NANDチップナンバレジスタ201にセットされたチップ番号に対応するチップセレクト信号CS1~CS16の1つをアクティブにする(ステップB31)。このチップセレクト信号はリード対象チップのチップイネーブル入力CEに供給され、これによってリード対象チップがイネーブル状態に設定される。

【0167】次に、ファームウェアは、NANDコマンドラッチイネーブル制御レジスタ204を1/Oライトして、そこにコマンドラッチイネーブル(CLE)信号をアクティブにすることを指定するCLEオンコマンドをセットする(ステップA34)。

【0168】NANDバスインタフェース19は、CLEオンコマンドのセットにตอบสนองして、コマンドラッチイネーブル信号CLEをアクティブ状態、つまり“H”レベルに設定する(ステップB32)。

【0169】次いで、ファームウェアは、NANDデータリード/ライトレジスタ202を1/Oライトして、そこにリードコマンド“00h”をセットする(ステップA35)。NANDバスインタフェース19は、リードコマンド“00h”のセットにตอบสนองして、ライトイネーブル信号WEをアクティブ状態、つまり“L”レベルに設定し、且つリードコマンド“00h”を入出力端子1/Oに転送する(ステップB33)。

【0170】ファームウェアは、NANDアドレスラッチイネーブル制御レジスタ205を1/Oライトして、そこにALEオンコマンドをセットする(ステップA36)。NANDバスインタフェース19は、ALEオンコマンドのセットにตอบสนองして、アドレスラッチイネーブル信号ALEをアクティブ状態、つまり“H”レベルに設定する(ステップB34)。

【0171】次いで、ファームウェアは、NANDデータリード/ライトレジスタ202を1/Oライトして、そこにメモリアドレス(カラムアドレスA7~A0、およびページアドレスA20~A8)をセットする(ステップA37)。NANDバスインタフェース19は、アドレスのセットにตอบสนองして、ライトイネーブル信号WEをアクティブ状態、つまり“L”レベルに設定し、且つセットされたメモリアドレスを入出力端子1/Oに転送する(ステップB35)。

【0172】このメモリアドレスが転送されると、リー

ド対象のフラッシュEEPROMにおいてはページリード動作が実行される。ページリード動作では、メモリアドレスで指定されたページに格納されている256バイトのユーザデータ、2バイトのLWCデータ、および8バイトのECCデータが読み出され、それがデータレジスタに転送される。このページリード動作期間中は、そのチップからのレディー/ビジー信号(RDY/BSY)はビジー状態を示し、ページリード動作が完了するとビジー状態からレディー状態に戻る。

【0173】ファームウェアは、リードアクセスしたチップに対応するNANDチップレディー/ビジーレジスタの内容によってページリード動作の完了を確認すると(ステップA38)、ECCチェック開始コマンドをECCスタート制御レジスタ206にセットし(ステップA39)、その後、ムーブストリング命令を実行して、データバッファ131からデータバッファ131への256バイトのユーザデータ転送を開始する(ステップA40)。

【0174】このデータ転送においては、NANDバスインタフェース19から出力されるリードイネーブル信号REに同期して、フラッシュEEPROMのデータレジスタに保持されている256+8バイトのデータが1バイト単位でシリアルに読み出される。

【0175】NANDバスインタフェース19は、フラッシュEEPROMから読み出されるデータに対して誤り検出のためのチェック演算を行い(ステップB36)、そのチェック結果をECCデータレジスタ209~214にセットする(ステップB37)。

【0176】ファームウェアは、ECCチェック演算の終了を確認すると、ECCデータレジスタ209~214の内容を参照して正常なデータをリードできたか否かを調べる(ステップA41)。もしリードデータにエラーが発生していることが検出された場合は、ファームウェアは、ECCデータレジスタ209~214にセットされているECCチェック演算結果を用いて、データバッファ131に転送されている256バイトのユーザデータの誤り訂正をデータバッファ131上で実行する。

【0177】このように、リードサイクルにおいては、フラッシュEEPROMからデータバッファ131へのデータ転送時にECCチェックがNANDバスインタフェース19によって自動的に実行される。

【0178】ところで、フラッシュEEPROM11-1~11-16の各々は、ページインクリメント機能を有している。このページインクリメント機能は、データレジスタからの1ページ分のデータ読み出しが終了した時に、自動的に次ページのページリードを開始する機能である。このため、前述した先頭ページのシリアルデータ読み出しが終了すると、フラッシュEEPROMは再びビジー状態となって、2ページ目のページリードを開始する。

【0179】したがって、2ページ目以降のデータ読み出しについては、ファームウェアおよびNANDバスインタフェース19は、前述のステップA31～A37、B31～B35のアドレスサイクルを実行する必要はない。それぞれ前述のステップA38～A41、B36、B37を繰り返すだけで良い。

【0180】次に、第2リードサイクルによって、冗長領域のデータだけを読み出す場合の動作について説明する。

【0181】この第2リードサイクルを実行する場合に必要なファームウェアおよびNANDバスインタフェース19の動作は、第1リードサイクルの場合とほとんど同じであり、違いは次の2点である。

【0182】(1) 図16のステップA35において、ファームウェアは、コマンド“00h”の代わりにコマンド“50h”を発行する。

【0183】(2) データレジスタからの2バイトのデータ読み出しでリードサイクルのシーケンスが終了する。

【0184】このため、LWCデータは第2リードサイクルを1回行うだけで読みだすことができるが、ECCデータを読み出す場合には、第2リードサイクルを3回繰り返すことが必要となる。

【0185】なお、第2リードサイクルにおいて、データレジスタのどの位置からデータを読み出すかは、アドレスサイクルで設定するカラムアドレスの値によって決定する事ができる。

【0186】次に、図18および図19を参照して、フラッシュEEPROMのブロックイレースサイクルの制御に必要なファームウェアの処理とNANDバスインタフェース19の動作について説明する。

【0187】図18はブロックイレースサイクルにおけるファームウェアの処理とNANDバスインタフェース19の動作を説明するフローチャートであり、図19はブロックイレースサイクルにおけるフラッシュEEPROMの動作を示すタイミングチャートである。

【0188】ここでは、ある特定の1ブロックの記憶内容を消去する場合を例にとって説明する。

【0189】マイクロプロセッサ4によって実行されるファームウェアは、まず、イレース対象ブロックを含むフラッシュEEPROMチップに対応するNANDチップレディ/ビジーレジスタ215-1～215-16の1つを参照して、そのイレース対象のフラッシュEEPROMチップがレディステート(待機中)か否かを確認する(ステップA51)。レディステートであることを確認すると、ファームウェアは、NANDチップナンバレジスタ201をI/Oライトして、そこにイレース対象フラッシュEEPROMチップに対応するチップ番号をセットする(ステップA52)。

【0190】次いで、ファームウェアは、NANDチ

プイネーブル制御レジスタ203をI/Oライトして、そこにチップイネーブル入力(CE)をアクティブにすることを指定するCEオンコマンドをセットする(ステップA53)。

【0191】この時、NANDバスインタフェース19は、NANDチップナンバレジスタ201にセットされたチップ番号に対応するチップセレクト信号CS1～CS16の1つをアクティブにする(ステップB51)。このチップセレクト信号はイレース対象チップのチップイネーブル入力CEに供給され、これによってイレース対象チップがイネーブル状態に設定される。

【0192】次に、ファームウェアは、NANDコマンドラッチイネーブル制御レジスタ204をI/Oライトして、そこにコマンドラッチイネーブル(CLE)信号をアクティブにすることを指定するCLEオンコマンドをセットする(ステップA54)。

【0193】NANDバスインタフェース19は、CLEオンコマンドのセットにตอบสนองして、コマンドラッチイネーブル信号CLEをアクティブ状態、つまり“H”レベルに設定する(ステップB52)。

【0194】次いで、ファームウェアは、NANDデータリード/ライトレジスタ202をI/Oライトして、そこにイレースコマンド“60h”をセットする(ステップA55)。NANDバスインタフェース19は、イレースコマンド“60h”のセットにตอบสนองして、ライトイネーブル信号WEをアクティブ状態、つまり“L”レベルに設定し、且つイレースコマンド“60h”を出力端子I/Oに転送する(ステップB53)。

【0195】ファームウェアは、NANDアドレスラッチイネーブル制御レジスタ205をI/Oライトして、そこにALEオンコマンドをセットする(ステップA56)。NANDバスインタフェース19は、ALEオンコマンドのセットにตอบสนองして、アドレスラッチイネーブル信号ALEをアクティブ状態、つまり“H”レベルに設定する(ステップB54)。

【0196】次いで、ファームウェアは、NANDデータリード/ライトレジスタ202をI/Oライトして、そこにメモリアドレス(ブロックアドレスA20 A12)をセットする(ステップA57)。NANDバスインタフェース19は、アドレスのセットにตอบสนองして、ライトイネーブル信号WEをアクティブ状態、つまり“L”レベルに設定し、且つセットされたメモリアドレスを出力端子I/Oに転送する(ステップB55)。

【0197】この後、ファームウェアは、CLEオンコマンドを発行した後に、NANDデータリード/ライトレジスタ202をI/Oライトして、そこにイレース開始コマンド“D0h”をセットする(ステップA58)。NANDバスインタフェース19は、まず、コマンドラッチイネーブル信号CLEをアクティブ状態に設定してからライトイネーブル信号WEをアクティブ状

態に設定し、且つイレース開始コマンド“D0h”を入出力端子I/Oに転送する(ステップB56)。

【0198】このイレース開始コマンド“D0h”が転送されると、イレース対象のフラッシュEEPROMにおいてブロックイレース動作およびそのベリファイ動作が自動実行される。

【0199】ブロックイレース動作では、ブロックアドレスで指定されたブロックの記憶内容だけが消去される。このブロックイレース動作期間中そのチップからのレディ/ビジー信号(RDY/BSY)はビジー状態を示し、ブロックイレース動作が完了するとビジー状態からレディ状態に戻る。

【0200】ファームウェアは、イレースしたチップに対応するNANDチップレディ/ビジーレジスタの内容によってブロックイレース動作の完了を確認すると(ステップA59)、NANDデータリード/ライトレジスタ202をI/Oライトして、そこにステータスリードコマンド“70h”をセットする(ステップA60)。NANDバスインタフェース19は、ステータスリードコマンド“70h”のセットに応答して、ライトイネーブル信号WE⁻をアクティブ状態、つまり“L”レベルに設定し、且つステータスリードコマンド“70h”を入出力端子I/Oに転送し、その後、リードイネーブル信号RE⁻をアクティブ状態、つまり“L”レベルに設定する(ステップB57)。

【0201】ステータスリードコマンド“70h”が転送されると、フラッシュEEPROMからはブロックイレース動作が成功したか否かを示すステータスデータが読み出され、それがNANDデータリード/ライトレジスタ202にセットされる。

【0202】ファームウェアは、NANDデータリード/ライトレジスタ202のステータスデータを参照してブロックイレース動作が成功したか否かを確認し、成功していればイレースサイクルを終了する(ステップA61)。もし失敗した場合には、代替ブロックとの代替処理などを行う。

【0203】なお、ブロックイレース開始コマンド“D0h”の発行前にステップA54～A57を繰り返し実行して複数のブロックアドレスを設定すれば、複数ブロックのイレースを順次行うマルチブロック動作を実行することもできる。

【0204】次に、図20のフローチャートを参照して、ホストシステムからライトコマンドが発行された際に実行される図1の半導体ディスク装置10全体のライト制御動作を説明する。このライト制御動作は、前述した各種の管理情報テーブルを利用する事によって行われる。

【0205】ホストシステムが外部バスインタフェース17のコマンドレジスタ176にライトコマンドをセットすると、マイクロプロセッサ14によって実行される

ファームウェアは、まず、そのライトコマンドを受け付ける(ステップA71)。次いで、ファームウェアは、ホストシステムからデータバッファ131へのライトデータの転送を開始する(ステップA72)。このデータ転送は、外部バスインタフェース17のデータレジスタ173を介して実行される。

【0206】ホストシステムによる半導体ディスク装置10のアクセスはセクタ単位で行われるので、データバッファ131には少なくとも1セクタ分のライトデータが格納される。

【0207】次いで、ファームウェアは、ホストシステムによって指定されたディスクアクセスのためのホストアドレス、つまり、セクタ番号、シリンダ番号、およびヘッド番号をセクタナンバレジスタ171、シリンダレジスタ174、およびドライブ/ヘッドレジスタ175からそれぞれ読取り、それをフラッシュEEPROM11-1～11-16をアクセスするためのアドレスに変換する(ステップA73)。このアドレス変換処理においては、図12で説明したように、アドレス変換テーブル132が利用される。このアドレス変換テーブル132のアドレス変換情報は、装置10の起動時において、フラッシュEEPROM11-1～11-16それぞれからリードしたアドレス変換テーブルのアドレス変換情報に基づいて構築されたものである。

【0208】アドレス変換処理の結果、ライトアクセス対象のチップ番号、およびライトアクセス対象チップ内のメモリアドレス(A20-A0)が求められる。

【0209】次に、ファームウェアは、メモリアドレス(A20-A0)の上位9ビット(A20-A12)で指定されるライトアクセス対象のブロック(以下、ライトブロックと称する)に対する1ブロック分のライトデータを作成するための処理(ステップA74～A78)に進む。

【0210】ここでは、ファームウェアは、まず、ライトブロック用のヘッダテーブルエリアをRAM13上に確保する(ステップA74)。次に、ファームウェアは、ライトブロックの先頭ページからLWCデータをリードする(ステップA75)。このLWCデータのリード処理では、前述したコマンド“50h”を用いた第2リードサイクルが実行され、これによってライトブロックの先頭ページからLWCデータだけが読み出される。このLWCデータは、RAM13の空きエリアまたはデータバッファ131に格納される。

【0211】この後、ファームウェアは、ライトブロックに“巻き添え消去”データが存在するか否かを調べる(ステップA76)。ここで、“巻き添え消去”データとはライトブロックに存在するデータの中でホストシステムによって書換が要求されていないデータを意味する。このため、ライトブロック内の全てのページに対する書込みが要求された場合には“巻き添え消去”データは存

庄しないか、ライトブロック内の途中のページから書き込みが開始される場合や、ライトデータのデータレングスが1ブロック(8セクタ)よりも少ない場合などには“巻き添え消去”データが存在する事になる。

【0212】“巻き添え消去”データが存在する場合に、ファームウェアは、コマンド“00h”を用いた第1リードサイクルを実行することによって、ライトブロックから“巻き添え消去”データを含むページをリードする(ステップA77)。この“巻き添え消去”データを含むページは、RAM13の空きエリアやデータバッファ13に格納される。

【0213】次いで、ファームウェアは、リードした“巻き添え消去”データのRAM13上の格納位置を示すRAMアドレスと、ホストシステムからのライトデータが格納されているRAM13上の格納位置を示すRAMアドレスとに基づいて、ライトブロックに書き込むべき1ブロック分のブロックデータに対応するヘッダテーブルを作成する。

【0214】すなわち、ファームウェアは、実際にデータを結合して1ブロック分のブロックデータを生成するのではなく、“巻き添え消去”データのRAM13上のアドレスとライトデータのデータバッファ131上のアドレスとをページ番号順に整理して保持するヘッダテーブル133を作成し、これによって仮想的なブロックデータを構築する(ステップA78)。

【0215】次いで、ファームウェアは、ライトブロックからリードしたLWCデータの値を+1カウントアップすることにより、そのLWCデータの値を更新する(ステップA79)。この時、もしLWCデータの値に桁上り(キャリー)が発生しなければ、ファームウェアは、ライトブロックに対するライトアクセス処理を実行する(ステップA80、A81)。

【0216】このライトアクセス処理では、まず、ライトブロックに対して図18で説明したブロックイレースサイクルが実行されて、ライトブロックの内容が全て消去される。次いで、ライトブロックの全ページに対して図14で説明したプログラムサイクルが実行される。このプログラムサイクルでは、ファームウェアは、まず、更新したLWCデータをNANDバスインタフェース19のLWCレジスタ208にセットし、次いで、ヘッダテーブル133に保持されているアドレス順にRAM13から256バイト単位で1ブロック分のデータを読み出し、それをフラッシュEEPROMに転送する。このように、ヘッダテーブル133を利用することにより、ファームウェアはライトデータと“巻き添え消去”データとを区別することなくデータ転送を行う事ができる。

【0217】転送される各256バイトのデータにはNANDバスインタフェース19によって自動的にECCデータが付加され、それがライト対象フラッシュEEPROMのライトブロックの対応するページに書き込まれ

る。

【0218】一方、LWCデータの値に桁上りが発生した場合には、ファームウェアは、スワッピング操作を実行するか否かを検討するために、以下の処理を行う。

【0219】すなわち、ファームウェアは、まず、第1リードサイクルを実行することによって、ライト対象のフラッシュEEPROMからMBIテーブルブロックをリードする(ステップA82)。リードされたMBIテーブルブロックの内容は、RAM13の空きエリアに格納される。

【0220】次いで、ファームウェアは、リードしたMBIテーブルブロックを検索してライトブロックに対応するUWCデータを検出し、そのUWCデータの値を更新、つまり+1カウントアップする(ステップA83)。MBIテーブルブロックにおいては、図8で説明したように、UWCデータは物理ブロックアドレス順に整理されて保持されている。このため、ファームウェアは、ライトブロックを示す物理ブロックアドレスをインデックスとして利用する事により、ライトブロックに対応するUWCデータを容易に見つける事ができる。

【0221】更新されたUWCデータの値は、MBIテーブルブロックと書換回数管理テーブル134双方に反映される。

【0222】この後、ファームウェアは、更新されたライトブロックのUWCデータの値をライト対象チップの最小UWCデータの値、および他の各チップの最小UWCデータの値と、順次比較する(ステップA84)。

【0223】ライト対象チップの最小UWCデータの値としては、ライト対象チップからリードしたMBIテーブルブロック内のスワップオーダーテーブルに保持されている最小UWCデータの値が利用される。これにより、RAM13上の書換回数管理テーブル134をUWCデータの小さい順にソーティングするといった処理を行うことなく、ライト対象チップの最小UWCデータの値を容易に知るができる。

【0224】また、他の各チップの最小UWCデータの値としては、RAM13上の書換回数管理テーブル134の値が利用される。この書換回数管理テーブル134の値を利用することにより、ライト対象チップ以外の他の全てのチップそれぞれからMBIテーブルブロックを順次リードする必要が無くなり、効率の良いスワッピング検討処理が実現される。

【0225】次いで、ステップA84の比較結果に基づき、ファームウェアは、ライトブロックの更新されたUWCデータとライト対象チップの最小UWCデータとの値の差が予め決められたしきい値(α)以上であるかを調べ(ステップA85)、その差が α 以上であれば、ライト対象チップ内でのスワッピング操作を行う(ステップA86)。

【0226】このチップ内スワッピング操作において

10

20

30

40

50

は、ファームウェアは、まず、ライト対象チップから被スワッピング対象のブロックをリードする。この場合、被スワッピング対象のブロックは最小UWCの値を持つブロックであり、このブロックの物理ブロックアドレスは、ステップA82でリードしたMBIテーブルブロックに含まれているスワップオーダーテーブル上の被スワッピング優先度NO. 1ブロック番号によって指定される。したがって、ファームウェアは、UWCデータのソーティングを行うことなく、被スワッピング対象のブロックを簡単に決定でき、またその被スワッピング対象のブロックの物理ブロックアドレスを認識する事ができる。

【0227】次に、ファームウェアは、ライトブロックと被スワッピング対象ブロック間でホストアドレスの入れ替えを行う。この場合、RAM132上のアドレス変換テーブル132とライト対象チップからリードしたMBIテーブルブロック双方のアドレス変換情報が更新される。

【0228】次に、ファームウェアは、ライト対象フラッシュEEPROMにおけるライトブロックと被スワッピング対象ブロック間の記憶内容の入れ替えのために、RAM13上に吸い上げた被スワッピングブロックの内容をライト対象フラッシュEEPROMのライトブロックに書き込むためのライトアクセスと、フラッシュEEPROMの被スワッピングブロックにRAM13上の1ブロック分のライトデータを書き込むためのライトアクセスを行う。この後、ファームウェアは、ライト対象フラッシュEEPROMにおけるMBIテーブルブロックの更新のために、ライト対象フラッシュEEPROMにおけるMBIテーブルブロックにRAM13上の更新されたMBIテーブルブロックの内容を書き込むためのライトアクセスを行う。これら3回のライトアクセスは、それぞれステップA81で説明した場合と同様の形態で実行される。

【0229】次に、ステップA86においてライトブロックの更新されたUWCデータとライト対象チップの最小UWCデータとの値の差が α 未満であった場合の処理について説明する。

【0230】この場合、ファームウェアは、ライトブロックの更新されたUWCデータと他のいずれかのチップの最小UWCデータとの値の差が予め決められたしきい値($\alpha - \epsilon$)以上であるか否かを調べ(ステップA87)、その差が $\alpha - \epsilon$ 以上であれば、ライト対象チップとその最小UWCデータの値を持つチップとの間でのスワッピング操作を行う(ステップA88)。

【0231】このチップ間に互るスワッピング操作においては、ファームウェアは、まず、被スワッピング対象のブロックを持つチップからMBIテーブルブロックをRAM13上にリードし、そのMBIテーブルブロックに含まれているスワップオーダーテーブル上の被スワ

ッピング優先度NO. 1ブロック番号を認識する。

【0232】次いで、ファームウェアは、被スワッピング優先度NO. 1ブロック番号が示す物理ブロックアドレスを用いて、その被スワッピング対象チップから被スワッピング対象ブロックをリードする。

【0233】次に、ファームウェアは、ライトブロックと被スワッピング対象ブロック間でホストアドレスの入れ替えを行う。この場合、RAM132上のアドレス変換テーブル132と、ライト対象チップからリードしたMBIテーブルブロックと、被スワッピング対象チップからリードしたMBIテーブルブロックのそれぞれのアドレス変換情報が更新される。

【0234】次に、ファームウェアは、ライト対象フラッシュEEPROMにおけるライトブロックと被スワッピング対象チップの被スワッピング対象ブロック間の記憶内容の入れ替えのために、RAM13上に吸い上げた被スワッピングブロックの内容をライト対象フラッシュEEPROMのライトブロックに書き込むためのライトアクセスと、被スワッピング対象フラッシュEEPROMの被スワッピングブロックにRAM13上の1ブロック分のライトデータを書き込むためのライトアクセスを行う。この後、ファームウェアは、ライト対象フラッシュEEPROM被スワッピング対象フラッシュEEPROMそれぞれのMBIテーブルブロックを更新するために、ライト対象フラッシュEEPROMにおけるMBIテーブルブロックにRAM13上の更新されたMBIテーブルブロックの内容を書き込むためのライトアクセスと、被スワッピング対象フラッシュEEPROMにおけるMBIテーブルブロックにRAM13上の更新されたMBIテーブルブロックの内容を書き込むためのライトアクセスとを行う。これら4回のライトアクセスは、それぞれステップA81で説明した場合と同様の形態で実行される。

【0235】このように、スワッピング操作はチップ内、又はチップ間で実行され、チップ間スワッピングの場合はチップ内スワッピングの場合よりも、リードアクセス操作およびライトアクセス操作がそれぞれ1回ずつ多く必要となる。

【0236】しかしながら、この実施例では、スワッピングを行うか否かを決定するためのしきい値($\alpha + \beta$)は、チップ内のスワッピング操作を行う場合のしきい値(α)よりも大きな値に設定されているので、チップ内スワッピングに比べチップ間スワッピングの発生頻度を抑制する事ができる。よって、チップ間スワッピング操作の多発による性能低下を防止する事ができる。

【0237】ステップA85、およびA87で共に置き換え回数の差が所定のしきい値よりも小さい場合には、ファームウェアは、ステップA83で更新されたUWCデータを含むMBIテーブルブロックをライト対象のフラッシュEEPROMに書き込むためのライトアクセス

を行いくステップA89)、次いで、ライトブロックに対するライトアクセス処理を実行する。これらライトアクセスは、それぞれステップA81で説明した場合と同様の形態で実行される。

【0238】以上のように、LWCデータの桁上りが発生しない場合にはスワッピング処理の検討がなされずに直ぐにライトアクセスが行われるので、桁上りが発生した場合よりもデータ書込みに要する全体の時間を短縮する事ができる。

【0239】LWCデータの値はライトアクセスの度にその初期値から+1ずつカウントアップされ、そして1K回に達した時に桁上りが発生する。このため、ディスク製造時や出荷時においてフラッシュEEPROM11-11~11-16に含まれる全てのブロックに書き込むLWCデータの初期値としては、同一の値ではなく、それぞれ異なる離散的な値を利用することが好ましい。この場合、連続ブロックに対するライトアクセスが継続的に発生したとしても桁上りが発生する時期がブロック毎にずれるので、桁上りに起因するスワッピングの検討処理およびスワッピング処理が集中して発生するという事態を防止する事ができる。

【0240】以上説明したように、この実施例においては、ECC演算回路21によってページ毎にECCデータが生成され、そのECCデータがECC演算の対象となったデータ列と同一のページに書き込まれる。このためたとえフラッシュEEPROMの複数ページそれぞれの同一ヒット位置に不良セルが発生しても、ECC演算の対象となるデータ列に含まれる不良セルは1つだけとなる。したがって、データ回復能力の高い複雑なECC演算式を利用することなく通常の簡単なECC演算によってエラー検出および訂正を行うことが可能となり、記憶信頼性が高く、且つ高速アクセスが可能な半導体ディスク装置10が実現される。

【0241】また、各ブロックの先頭ページの冗長領域には、そのブロックに対応する書換回数情報としてLWCデータが格納されており、ライト対象ブロックに対するデータ書込みの際に、そのLWCデータの値が更新される。その更新されたLWCデータは、先頭ページに書き込むべきライトデータに付加されてフラッシュEEPROMのデータレジスタ112に転送される。このように、各ブロック内でLWCデータが管理されているので、同一ブロックに対するライトアクセスだけで、データ書込みとLWCデータの更新を行う事ができ、結果的にライトアクセス性能の向上を図る事ができる。

【0242】

【発明の効果】以上説明したように、この発明によれば、フラッシュEEPROMの各ページの冗長領域を利用してECCや書換回数情報を管理することによって、十分な記憶データの信頼性およびフラッシュEEPROMの高寿命化を実現することが可能となる。

【図面の簡単な説明】

【図1】この発明の一実施例に係る半導体ディスク装置の構成を示すブロック図。

【図2】図1の半導体ディスク装置において使用されるNAND型フラッシュEEPROMの構造を示す図。

【図3】図1の半導体ディスク装置に設けられているI/Oレジスタ群を示す図。

【図4】図1の半導体ディスク装置におけるメモリマップの一例を示す図。

10 【図5】図1の半導体ディスク装置に設けられたフラッシュEEPROM内の複数ブロックそれぞれの利用形態を示す図。

【図6】図5に示したブロックの中でユーザデータの格納に使用されるメモリブロックのデータ格納形式を示す図。

【図7】図5に示したブロックの中でメモリブロック管理情報の格納に使用されるMB1テーブルブロックのデータ格納形式を示す図。

【図8】図7のMB1テーブルブロックに格納されるアドレス変換およびUWCテーブルの内容を示す図。

20 【図9】図7のMB1テーブルブロックに格納されるスワップオーダーテーブルの内容を示す図。

【図10】図5に示したブロックの中で図1の半導体ディスク装置のコンヒグ情報の格納に使用されるコンヒグブロックのデータ格納形式を示す図。

【図11】図1の半導体ディスク装置のRAM上に設けられるアドレス変換テーブルの構成を示す図。

【図12】図11のアドレス変換テーブルを利用したホストアドレスからメモリアドレスへの変換処理の原理を説明するための図。

30 【図13】図1の半導体ディスク装置に設けられたECC演算回路の動作を説明するための図。

【図14】図1の半導体ディスク装置においてフラッシュEEPROMにデータを書き込む場合のファームウェアの処理とハードウェアの動作を説明するフローチャート。

【図15】図1の半導体ディスク装置に設けられたフラッシュEEPROMのプログラムサイクルを説明するタイミングチャート。

40 【図16】図1の半導体ディスク装置においてフラッシュEEPROMからデータを読み出す場合のファームウェアの処理とハードウェアの動作を説明するフローチャート。

【図17】図1の半導体ディスク装置に設けられたフラッシュEEPROMのシーケンシャルリードサイクルを説明するタイミングチャート。

50 【図18】図1の半導体ディスク装置においてフラッシュEEPROMに格納されたデータを消去する場合のファームウェアの処理とハードウェアの動作を説明するフローチャート。

【図19】図1の半導体ディスク装置に設けられたフラッシュEEPROMのブロックレージサイクルを説明するタイミングチャート。

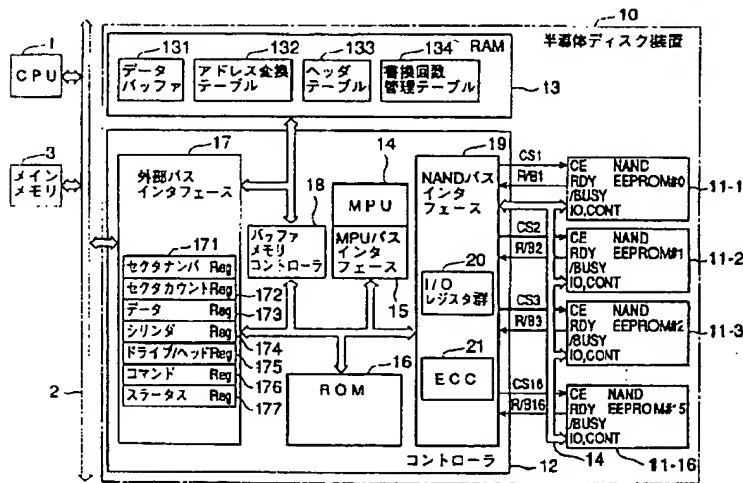
【図20】図1の半導体ディスク装置においてホストシステムからのライト要求を受信してからフラッシュEEPROMをライトアクセスするまでに実行されるファームウェアの処理を説明するフローチャート。

【図21】従来の半導体ディスク装置によるフラッシュEEPROMに対するデータ格納形式を示す図。

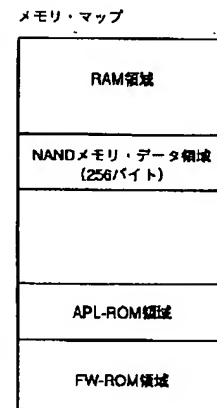
＊【符号の説明】

10…半導体ディスク装置、11-1～11-16…NAND型フラッシュEEPROM、12…コントローラ、13…RAM、14…マイクロプロセッサ、16…ROM、17…外部バスインタフェース、19…NANDバスインタフェース、21…ECC演算回路、131…データバッファ、132…アドレス交換テーブル、133…ヘッダテーブル、134…書換回数管理テーブル、131…データバッファ、132…アドレス交換テーブル、133…ヘッダテーブル、134…書換回数管理テーブル。

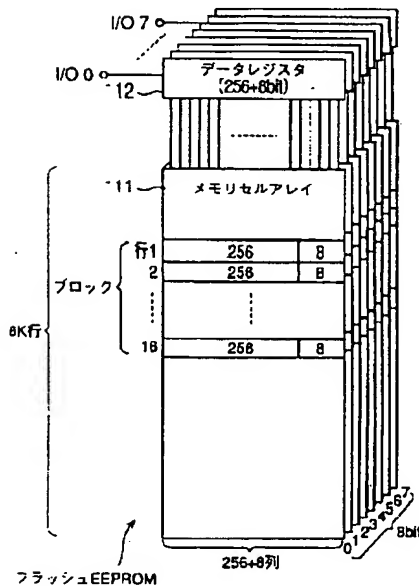
【図1】



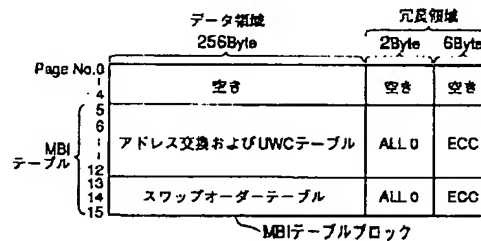
【図4】



【図2】



【図7】



【図3】

NANDバスインタフェースのI/Oレジスタ群

NAND チップNo.	Reg	201
NAND データリード/ライト	Reg	202
NAND CE制御	Reg	203
NAND CLE制御	Reg	204
NAND ALE制御	Reg	205
ECC スタート制御	Reg	206
ECC レディ/ビジーステータス	Reg	207
LWC	Reg	208
ECC データ No.0	Reg	209
ECC データ No.1	Reg	210
ECC データ No.2	Reg	211
ECC データ No.3	Reg	212
ECC データ No.4	Reg	213
ECC データ No.5	Reg	214
NAND チップ No.0 R/B	Reg	215-1
⋮		
NAND チップ No.15 R/B	Reg	215-16

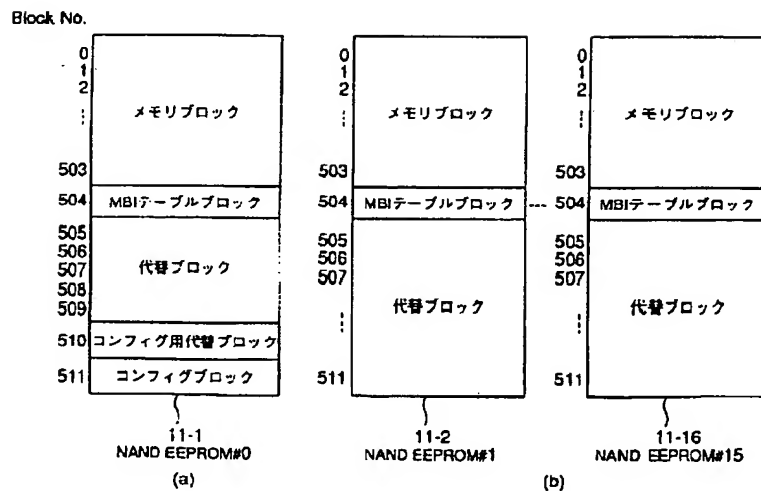
【図8】

MBIテーブル内
アドレス

001h,00h	物理ブロック番地0に対応するホスト番地	ページ5
003h,002h	物理ブロック番地0に対応するUWC	
005h,004h	物理ブロック番地1に対応するホスト番地	
007h,006h	物理ブロック番地1に対応するUWC	
	物理ブロック番地63に対応するホスト番地	ページ6
	物理ブロック番地63に対応するUWC	
	物理ブロック番地448に対応するホスト番地	ページ11
	物理ブロック番地448に対応するUWC	
	物理ブロック番地449に対応するホスト番地	ページ12
	物理ブロック番地449に対応するUWC	
7FDh,7FCh	物理ブロック番地511に対応するホスト番地	
7FFh,7FEh	物理ブロック番地511に対応するホスト番地	

アドレス変換およびUWCテーブル

【図5】



【図6】

メモリブロック	Page No.	データ領域 256Byte		冗長領域	
		2Byte		6Byte	
ブロック0 (4K+128/バイト)	0	ユーザデータ (セクタ0 1/2)	- W C	ECC	
ブロック1 (4K+128/バイト)	1	ユーザデータ (セクタ0 2/2)	BAD セクタ情報	ECC エラー情報	ECC
	2	ユーザデータ (セクタ1 1/2)	- 空 き -	ECC	
	3	ユーザデータ (セクタ1 2/2)	BAD セクタ情報	ECC エラー情報	ECC
	4	ユーザデータ (セクタ2 1/2)	- 空 き -	ECC	
	5	ユーザデータ (セクタ2 2/2)	BAD セクタ情報	ECC エラー情報	ECC
ブロック502 (4K+128/バイト)					
ブロック503 (4K+128/バイト)	14	ユーザデータ (セクタ7 1/2)	- 空 き -	ECC	
	15	ユーザデータ (セクタ7 2/2)	BAD セクタ情報	ECC エラー情報	ECC

11-1
NAND EEPROM#0

不良セル

ブロック0

【図9】

MBI テーブル内 アドレス	1Byte	
800h	最小UWC (H)	0-255/ 256-511 共通
801h	最小UWC (L)	
802h	最小UWCのブロック数	
803h	最小UWC+1のブロック数	
804h	最小UWC+2のブロック数	
...	...	0-255/ 256-511 共通
87Ah	最小UWC+120のブロック数	
87Bh	- reserved -	
87Ch	MBIテーブル自身の書き換え回数	
87Fh	被スワップ1st優先ブロックNo.	
880h	被スワップ2nd優先ブロックNo.	256-511 ブロックの オーダ テーブル2
881h	...	
97Fh	被スワップ255th優先ブロックNo.	
980h	最小UWC (H)	
981h	最小UWC (L)	
982h	最小UWCのブロック数	256-511 ブロックの オーダ テーブル2
983h	最小UWC+1のブロック数	
984h	最小UWC+2のブロック数	
...	...	
9FAh	最小UWC+120のブロック数	
9FBh	空き	256-511 ブロックの オーダ テーブル2
9FFh	被スワップ1st優先ブロックNo.	
A00h	被スワップ2nd優先ブロックNo.	
A01h	...	
AFh	被スワップ255th優先ブロックNo.	

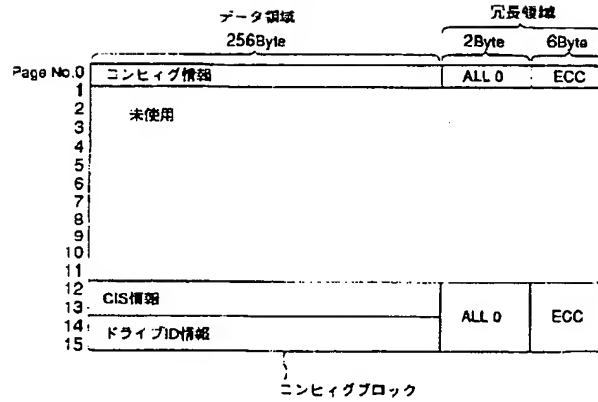
スワップオーダーテーブル

【図11】

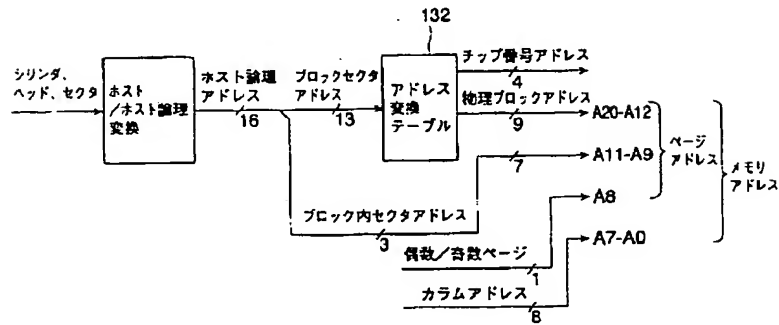
ホストアドレス [ブロックセクタ アドレス(13bit)]	物理アドレス (13bit)	
	チップ番号アドレス (4bit)	チップ内物理ブロック アドレス (9bit)
0		
1		
2		
3		
4		
5		
6		
7		
...
8K		

132 アドレス変換テーブル

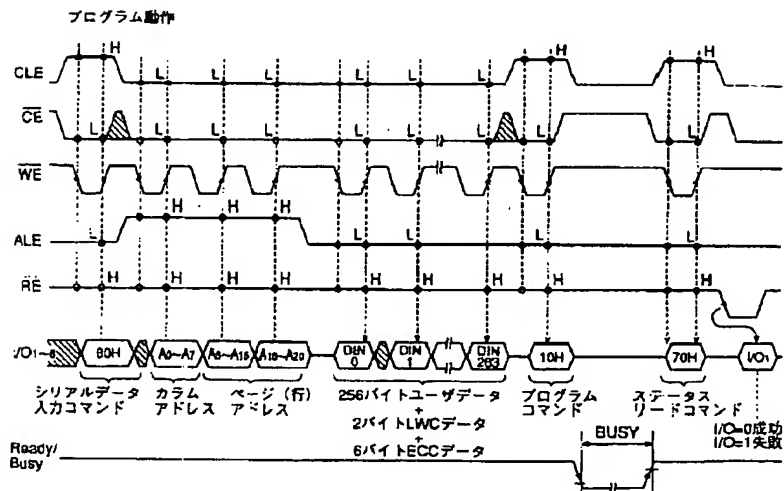
【図10】



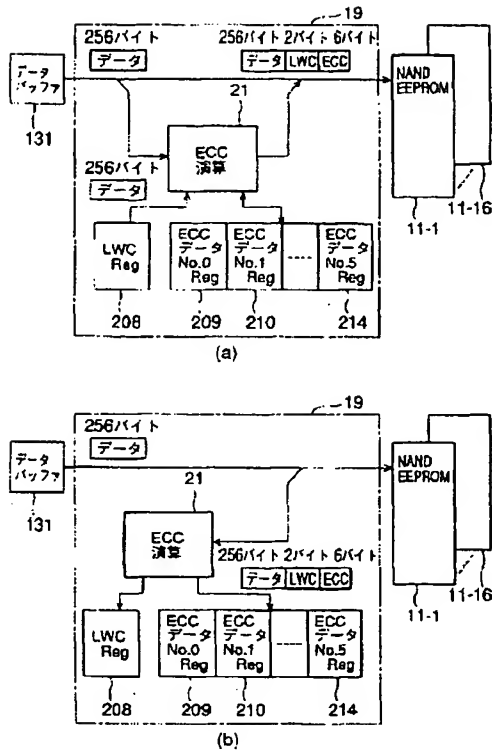
【図12】



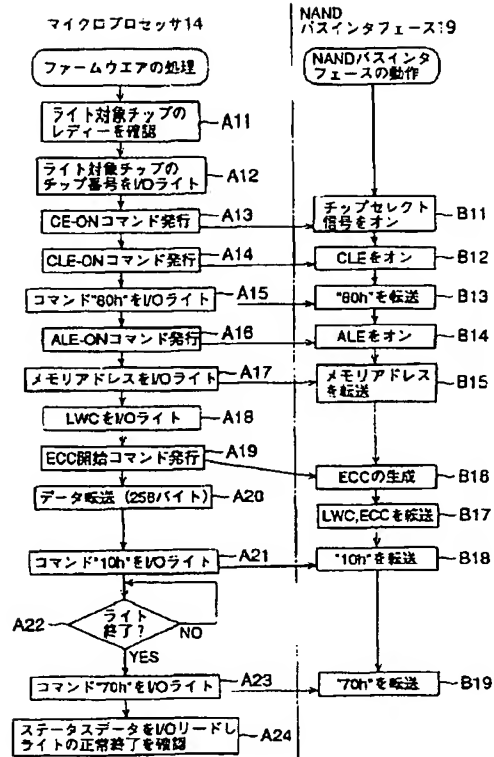
【図15】



【図13】

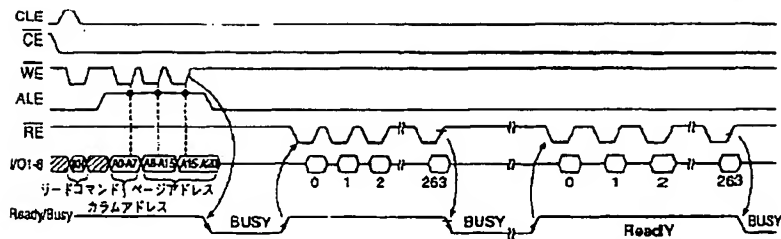


【図14】

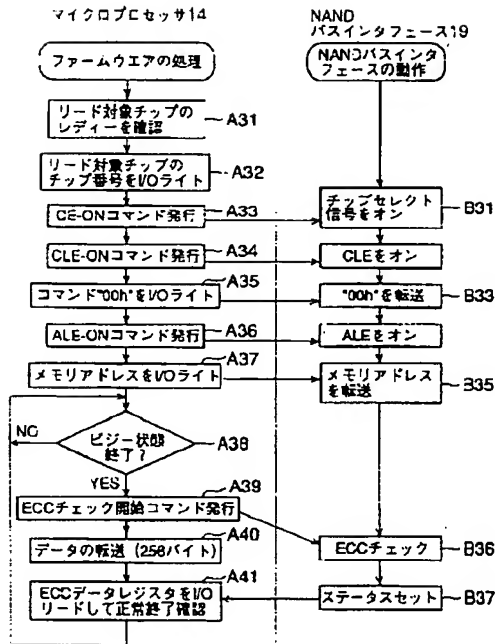


【図17】

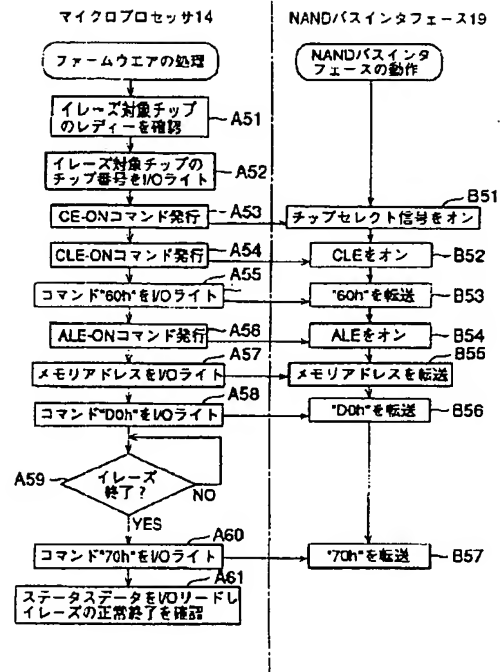
シーケンシャルリード動作



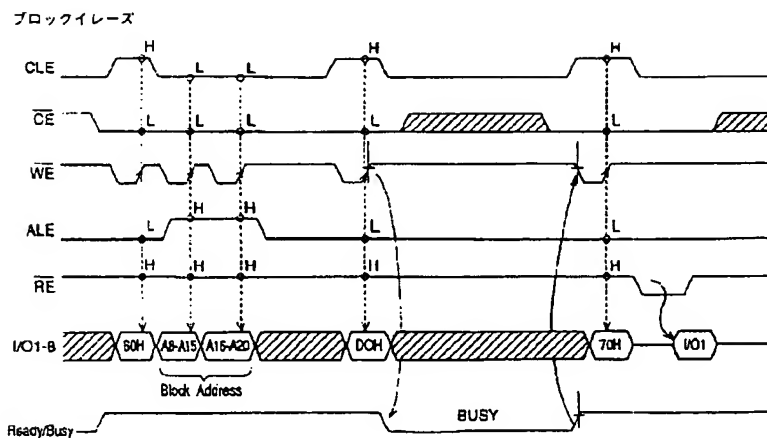
〔図16〕



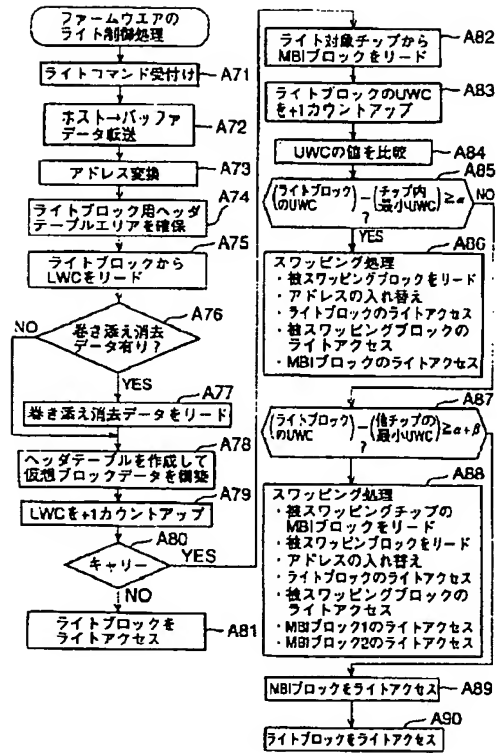
〔図18〕



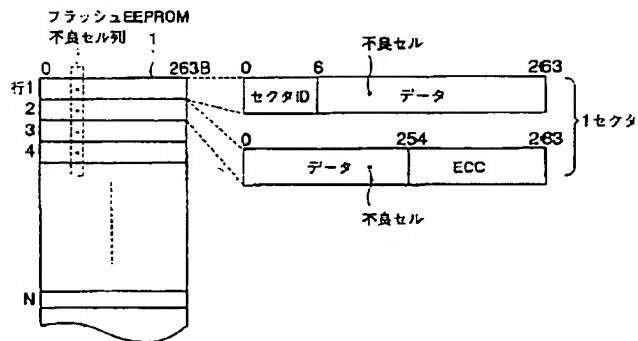
〔図19〕



【図20】



【図21】



* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The memory cell array which has two or more pages in which each includes a data storage area and a redundancy field, Have a data register holding the data for 1 page, and the flash EEPROM by which data transfer between a data register and a memory cell array is performed per page is built in. In the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment The data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, A light demand is answered from said host equipment, and the light access means which carries out light access of said flash EEPROM is provided. This light access means The light data stored in said data buffer are calculated in the data stream unit equivalent to the size of the data storage area of each of said page. So that it may be written in the data storage area and redundancy field an error correcting code generation means to generate an error correcting code for every data stream, and whose error correcting code corresponding to said data stream and it are the same pages Semi-conductor TISUKU equipment characterized by including a means to add the error correcting code corresponding to it for every data stream, and to transmit to the data register of said flash EEPROM.

[Claim 2] It is RAM disk equipment according to claim 1 characterized by answering a lead demand from said host equipment, providing further the read access means which carries out read access of said flash EEPROM, and for this read access means calculating the error correcting code of each page by which reading appearance is carried out from said flash EEPROM, and including an error correcting code operation means to perform error detection and correction for every data stream of each page.

[Claim 3] RAM disk equipment according to claim 1 characterized by providing further the means which writes the bad sector information which shows the bad sector specified from said host equipment in the redundancy field of the page corresponding to the bad sector.

[Claim 4] RAM disk equipment according to claim 1 characterized by providing further the means which writes the error information which shows the ECC error generating sector specified from said host equipment in the redundancy field of the page corresponding to the error generating sector.

[Claim 5] RAM disk equipment according to claim 1 characterized by being constituted so that ROM in which the program performed by said host equipment was stored may be provided further, the lead demand of the predetermined address from said host equipment may be answered and a program may be transmitted to said host equipment from said ROM.

[Claim 6] The memory cell array which has two or more pages in which each includes a data storage area and a redundancy field, Have a data register holding the data for 1 page, and the flash EEPROM by which data transfer between a data register and a memory cell array is performed per page is built in. In the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment A microprocessor and the data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, The flash memory interface circuitry which is controlled by said microprocessor and carries out read/write access of said flash EEPROM, The I/O register group in which read/write is possible is provided by said microprocessor. Said flash memory interface circuitry The light data by which reading appearance is carried out in the data stream unit which is equivalent to the size of the data storage area of each of said page from said data buffer with said microprocessor are calculated. So that it may be written in the error correcting code generation circuit which generates an error correcting code for every data stream, and the data storage area and redundancy field whose error correcting code corresponding to said data stream and it is the same page The light data transfer circuit which adds the error correcting code corresponding to it for every data stream, and is transmitted to the data register of said flash EEPROM, The lead data transfer circuit which takes out the data stream stored in said data storage area from the lead data by which reading appearance is carried out from the data register of said flash EEPROM, and is transmitted to said data buffer, Error detection of said lead data is performed using the error correcting code contained in said lead data. The error correcting code checking circuit which sets the detection result to said I/O register is included. Said microprocessor RAM disk equipment characterized by performing the error correction of the lead data transmitted to said data buffer according to the result of an operation set to said I/O register.

[Claim 7] The memory cell array in which the count information of rewriting which shows the count of rewriting of the block in the redundancy field of the predetermined page of each block is stored including two or more blocks which have two or more pages in which each includes a data storage area and a redundancy field, respectively, Have a data register holding the data for 1 page, and data transfer between a data register and a memory cell array is performed per page. Elimination

actuation builds in the flash EEPROM performed per block. It is RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment. The data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, A light demand is answered from said host equipment, and the light access means which carries out light access of said flash EEPROM is provided. This light access means A means to read the count information of rewriting from said predetermined page within the block for a light specified by said light demand, and to update the value of the count information of rewriting, Add said updated count information of rewriting to the light data which should be written in said predetermined page, and page data are generated. RAM disk equipment characterized by providing a means to transmit said page data to the data register of said flash EEPROM so that it may be stored in the data storage area and redundancy field said light data and said whose updated count information of rewriting are said predetermined pages.

[Claim 8] The memory cell array in which the count information of rewriting which shows the count of rewriting of the block in the redundancy field of the predetermined page of each block is stored including two or more blocks which have two or more pages in which each includes a data storage area and a redundancy field, respectively, Have a data register holding the data for 1 page, and a command is answered from the outside. The elimination actuation which eliminates the content of storage per block, the light actuation which writes data in a memory cell array per page, And the flash EEPROM which performs lead actuation which reads data from a memory cell array in a page unit is built in. In the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment The data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, A light demand is answered from said host equipment, and the light access means which carries out light access of said flash EEPROM is provided. This light access means Publish a lead command and said flash EEPROM is set as a lead mode of operation. A means to read the count information of rewriting from said predetermined page within the block for a light specified by said light demand, and to update the value of the count information of rewriting, When the page of the outside for light access exists in the block for a light, The data stored in the page of the outside for [the] light access are read on said data buffer. The storing location on the data buffer of each data which constitutes the block data for 1 block which should be written in the block for [said] a light based on the address on the data buffer of the read data, and the address of the light data stored in said data buffer A means to generate the header table holding the shown address information, and a means to publish an erasion command, to set said flash EEPROM as an elimination mode of operation, and to eliminate the content of storage of said block for a light, With reference to the address information of said header table, the light data which should be written in said predetermined page are read from said data buffer. So that said updated count information of rewriting may be added to the light data, page data may be generated and it may be stored in the data storage area and redundancy field said light data and said whose updated count information of rewriting are said predetermined pages A means to transmit said page data to the data register of said flash EEPROM, RAM disk equipment characterized by including a means to read the light data which should be written in in addition to said predetermined page from said data buffer one by one with reference to the address information of said header table, and to transmit it to the data register of said flash EEPROM.

[Claim 9] The memory cell array in which the count information of rewriting which shows the count of rewriting of the block in the redundancy field of the predetermined page of each block is stored including two or more blocks which have two or more pages in which each includes a data storage area and a redundancy field, respectively, Have a data register holding the data for 1 page, and a command is answered from the outside. The elimination actuation which eliminates the content of storage per block, the light actuation which writes data in a memory cell array per page, And the flash EEPROM which performs lead actuation which reads data from a memory cell array in a page unit is built in. In the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment The data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, A light demand is answered from said host equipment, and the light access means which carries out light access of said flash EEPROM is provided. This light access means Publish a lead command and said flash EEPROM is set as a lead mode of operation. A means to read the count information of rewriting from said predetermined page within the block for a light specified by said light demand, and to update the value of the count information of rewriting, When the page of the outside for light access exists in the block for a light, The data stored in the page of the outside for [the] light access are read on said data buffer. The block data for 1 block which should be written in the block for [said] a light based on the address on said data buffer of the read data, and the address on said data buffer of the light data stored in said data buffer A means to generate the header table holding the address information which shows the storing location on the data buffer of each data to constitute, A means to publish an erasion command, to set said flash EEPROM as an elimination mode of operation, and to eliminate the content of storage of said block for a light, The means which reads said block data from said data buffer one by one with reference to the address information of said header table in the data stream unit equivalent to the size of said data storage area, An error correcting code generation means by which reading appearance is carried out from said data buffer to generate an error correcting code for every data stream, The error correcting code corresponding to the data stream to the data stream which should be written in said predetermined page, And so that said updated count information of rewriting may be added, page data may be generated and it may be written in the redundancy field the data storage area said whose data stream is said predetermined page, said updated count information of rewriting, and whose error correcting code are said predetermined pages A means to transmit said generated page data to the data register of said flash EEPROM, Add the error correcting code corresponding to the data stream to the data stream which should be written in the

page for every pages of other other than said predetermined page, and page data are generated. RAM disk equipment characterized by including a means to transmit said generated page data to the data register of said flash EEPROM so that it may be written in the data storage area and redundancy field said whose data stream and error correcting code are each page. [Claim 10] Two or more blocks which have two or more pages in which each includes a data storage area and a redundancy field, respectively are included. The 1st count data of rewriting which consists of the high-order-bit sections of the count data of rewriting in which the count of rewriting of each block is shown is stored in a predetermined block. The memory cell array by which the 2nd count data of rewriting which consists of the lower bit sections of the count data of rewriting corresponding to the block is stored in the redundancy field of the predetermined page of each block, Have a data register holding the data for 1 page, and a command is answered from the outside. The elimination actuation which eliminates the content of storage per block, the light actuation which writes data in a memory cell array per page, And the flash EEPROM which performs lead actuation which reads data from a memory cell array in a page unit is built in. In the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment The data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, Publish a lead command, set said flash EEPROM as a lead mode of operation, and the 1st count data of rewriting is read from said predetermined block. The count management tool of rewriting which manages the count of rewriting of said flash EEPROM for every block according to the 1st count data of rewriting, A light demand is answered from said host equipment, and the light access means which carries out light access of said flash EEPROM is provided. This light access means Publish a lead command and said flash EEPROM is set as a lead mode of operation. A means to read the 2nd count data of rewriting from said predetermined page within the block for a light specified by said light demand, and to update the value of the 2nd count data of rewriting, The data stored in the page of the outside for [within the block for a light] light access are read on said data buffer. The storing location on the data buffer of each data which constitutes the block data for 1 block which should be written in the block for [said] a light based on the address on the data buffer of the read data, and the address of the light data stored in said data buffer A means to generate the header table holding the shown address information, and a means to publish an erasion command, to set said flash EEPROM as an elimination mode of operation, and to eliminate the content of storage of said block for a light, With reference to the address information of said header table, a means to carry out reading appearance of said block data in a page unit from said data buffer, and to transmit to the data register of said flash EEPROM is included. When the digit riser from the count data of rewriting to said 2nd count data [1st] of rewriting generates said count management tool of rewriting by renewal of said 2nd count data of rewriting, RAM disk equipment characterized by including a means to update the value of the 1st count data of rewriting corresponding to said block for a light.

[Claim 11] Said count management tool of rewriting is RAM disk equipment according to claim 10 with which the difference of the count of rewriting with the block for a light is characterized by to provide further a means determine the block which reaches beyond a predetermined value as a swapping point block, and a swapping means replace mutually the block address and the content of storage of said block for a light, and said swapping point block when the value of the 1st count data of rewriting of said block for a light is updated.

[Claim 12] RAM disk equipment according to claim 11 characterized by said thing [that two or more respectively discrete values are stored in the redundancy field of the predetermined page of a block as initial value of said 2nd count data of rewriting].

[Claim 13] In said predetermined block, said count information of the minimum rewriting which shows two or more values of the 1st minimum count data of rewriting in a block, The swapping table holding the priority information which shows the priority of a swapping point block according to the value of said 1st count data of rewriting is stored. Said count management tool of rewriting When the value of the 1st count data of rewriting of said block for a light is updated, A means by which compare the value which the count information of the minimum rewriting on the count of rewriting of the block for a light and said swapping table shows, and the difference of the count of rewriting judges whether it is beyond a predetermined value, When the difference of the count of rewriting is beyond a predetermined value, with reference to said priority information on said swapping table, the swapping point block of whenever [of the highest priority] is determined. RAM disk equipment according to claim 10 characterized by providing further a swapping means to replace mutually the block address and the content of storage of said block for a light and swapping point block of whenever [said top priority].

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] Especially this invention builds in the flash EEPROM which has two or more page fields where each includes a data storage area and a redundancy field about RAM disk equipment, and relates to the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment.

[0002]

[Description of the Prior Art] Many of information processors, such as the conventional workstation and a personal computer, used the magnetic disk drive as secondary storage. While a magnetic disk drive has an advantage, like a bit unit price with the high dependability of record is cheap, it has a fault, such as being weak, in a physical impact with the great size of equipment.

[0003] Then, attentions have gathered for RAM disk equipment strong also against an impact with it in recent years. [the small size of equipment, and] [physical] The flash EEPROM whose RAM disk equipment is the semiconductor memory of the non-volatile in which package elimination is possible electrically is used as secondary storage, such as a personal computer, like the conventional magnetic disk drive etc. Since a part for mechanical moving part like the magnetic head of a magnetic disk drive or a revolution disk is not included in this RAM disk equipment, it is hard to generate malfunction and failure by the physical impact. Moreover, there is an advantage, like the size as equipment also becomes small.

[0004] By the way, in RAM disk equipment, the ECC calculation function is prepared, and ECC corresponding to the content of data is usually generated and added to the data written in a flash EEPROM. By using this ECC calculation function, the dependability of the data written in a flash EEPROM can be raised.

[0005] In the former which shows the typical data storage format adopted with conventional RAM disk equipment with an ECC calculation function function to drawing 21 , ECC is generated for every sector which is a disk-accessing unit by host equipment. For this reason, ECC is stored in the location which follows that sector data, when sector data with the size of 512 bytes are continued and stored in 2 pages of a flash EEPROM as shown in drawing 21 . Although determined by the physical size of a flash EEPROM, when in what page of a flash EEPROM sector data are continued and stored uses the 16M bit flash EEPROM with the largest size by which current development is carried out, 512 bytes of sector data will be continued and stored in 2 pages of a flash EEPROM 1 like a graphic display.

[0006] In this case, the sector data covering those 2 pages calculate, and ECC is generated. This generated ECC follows sector data and is stored in the 2nd page.

[0007] However, in the conventional RAM disk equipment which used such a data storage format, consideration of the following error generating properties which a flash EEPROM has produces the nonconformity of the dependability of data storage falling.

[0008] That is, in the semiconductor memory which makes a flash EEPROM the start, the error mode in which a defect occurs simultaneously exists in two or more memory cells connected not only to a certain specific memory cell but to the same bit line. This error mode is caused by the defect of a bit line, and the poor contact between a bit line and a cell.

[0009] When such error mode occurs, a defect cell is generated in the same bit position of each two or more pages of a flash EEPROM as shown in drawing 21 . In this case, in conventional RAM disk equipment, a two-place error will be generated like a graphic display at the data stream set as the object of an ECC operation.

[0010] Although detection and correction of the error can generally be easily performed by the error detection and correction using ECC when an error is one place, about two or more errors, the detection and correction become difficult. In order to correspond to such an error, the complicated high ECC operation expression of data bounce-back capability is needed.

[0011] However, if ECC operation expression is complicated, while the configuration of RAM disk equipment will be complicated, in order for generation of ECC at the time of the data light and the ECC check operation at the time of a data lead to take much time amount, the fault that the access rate of RAM disk equipment falls is caused.

[0012] Moreover, with conventional RAM disk equipment, all not only of generation of ECC and an ECC check operation but the error detection based on the result of an operation and error correction processing were performed by hardware. For this reason, there was a problem that the gate number for realizing that hardware increased, or a read/write rate fell as a result by complication of hardware.

[0013] Furthermore, with conventional RAM disk equipment, package management of the count of rewriting of each block

of a flash EEPROM is carried out by the count information of rewriting stored in the specific block of the flash EEPROM. For this reason, in case data are written in a flash EEPROM, two light accesses with light access for updating light access for data writing and the count information of rewriting are needed. Therefore, while the time amount which data writing takes became long, the count of light access to the block which has memorized the count information of rewriting projected and increased compared with other blocks, and there was a problem to which the life of a flash EEPROM becomes short by this.

[0014]

[Problem(s) to be Solved by the Invention] With conventional RAM disk equipment, the physical structure of a flash EEPROM was not fully taken into consideration. For this reason, even if it prepared the ECC calculation function, sufficient storage dependability was not able to be acquired. Moreover, since the count information of rewriting was concentrated and stored in the specific block, only the count of rewriting of the block increased and there was a fault to which the life of a flash EEPROM becomes short by this.

[0015] This invention was made in view of such a point, uses the redundancy field of each page of a flash EEPROM effectively, and enables it to manage ECC and the count information of rewriting, and it aims at offering the RAM disk equipment which can realize high life-ization of the dependability of enough stored data, and a flash EEPROM.

[0016]

[Means for Solving the Problem and its Function] The memory cell array in which invention concerning claim 1 has two or more pages in which each includes a data storage area and a redundancy field, Have a data register holding the data for 1 page, and the flash EEPROM by which data transfer between a data register and a memory cell array is performed per page is built in. In the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment The data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, A light demand is answered from said host equipment, and the light access means which carries out light access of said flash EEPROM is provided. This light access means The light data stored in said data buffer are calculated in the data stream unit equivalent to the size of the data storage area of each of said page. So that it may be written in the data storage area and redundancy field an error correcting code generation means to generate an error correcting code for every data stream, and whose error correcting code corresponding to said data stream and it are the same pages It is characterized by including a means to add the error correcting code corresponding to it for every data stream, and to transmit to the data register of said flash EEPROM.

[0017] In this RAM disk equipment, an error correcting code is generated for every page of a flash EEPROM, and it is written in the redundancy field that error correcting code of whose is each page. For this reason, even if, even if a two or more pages defect cel occurs in each same bit position, the defect cel contained in the data stream of a flash EEPROM set as the operation object of an error correcting code is set only to one. Therefore, without using the complicated high ECC operation expression of data bounce-back capability, by the easy usual ECC operation, it becomes possible to perform error detection and correction, and the RAM disk equipment in which rapid access is possible is realized highly [storage dependability].

[0018] Moreover, in invention concerning claim 3 and claim 4, it is characterized by writing not only an error correcting code but the bad sector information specified from host equipment and error information in a redundancy field, respectively.

[0019] By this, since it becomes unnecessary to carry out the centralized control of bad sector information or the error information, it becomes unnecessary to prepare the exclusive field for storing these information, and the data storage area of a flash EEPROM can be used effectively for storing of user data.

[0020] Moreover, invention concerning claim 5 is characterized by being constituted so that ROM in which the program performed by said host equipment was stored may be provided in addition to the configuration of claim 1, the lead demand of the predetermined address from said host equipment may be answered and a program may be transmitted to said host equipment from said ROM.

[0021] In this RAM disk equipment, ROM in which the program of an application program, an operating system, etc. other than a flash EEPROM was stored is prepared, and when the lead demand of a predetermined address is published from host equipment, reading appearance of the program of this ROM is carried out. Since rewriting does not generate programs, such as an application program, the data storage area of an expensive flash EEPROM can be used effectively for storing of user data by storing the program in comparatively cheap ROM.

[0022] Moreover, the memory cell array in which invention concerning claim 6 has two or more pages in which each includes a data storage area and a redundancy field, Have a data register holding the data for 1 page, and the flash EEPROM by which data transfer between a data register and a memory cell array is performed per page is built in. In the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment A microprocessor and the data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, The flash memory interface circuitry which is controlled by said microprocessor and carries out read/write access of said flash EEPROM, The I/O register group in which read/write is possible is provided by said microprocessor. Said flash memory interface circuitry The light data by which reading appearance is carried out in the data stream unit which is equivalent to the size of the data storage area of each of said page from said data buffer with said microprocessor are calculated. So that it may be written in the error correcting code generation circuit which generates an error correcting code for every data stream, and the data storage area and redundancy field whose error correcting code corresponding to said data stream and it is the same page The light data transfer circuit which adds the error correcting code corresponding to it for every data stream, and is transmitted to the data register of said

flash EEPROM, The lead data transfer circuit which takes out the data stream stored in said data storage area from the lead data by which reading appearance is carried out from the data register of said flash EEPROM, and is transmitted to said data buffer, Error detection of said lead data is performed using the error correcting code contained in said lead data. Said microprocessor is characterized by performing the error correction of the lead data transmitted to said data buffer according to the result of an operation set to said I/O register including the error correcting code checking circuit which sets the detection result to said I/O register.

[0023] The error correction of lead data is performed in this RAM disk equipment by the firmware which the check for generation of an error correcting code and error detection is performed by the hardware logic of a flash memory interface circuitry, and is performed by the microprocessor. Therefore, a load distribution is efficiently made by firmware and hardware and the configuration of hardware can be simplified rather than the case where all of the ECC check operation for generation of ECC and error detection and an error correction are performed by hardware.

[0024] Moreover, invention concerning claim 7 includes two or more blocks which have two or more pages in which each includes a data storage area and a redundancy field, respectively. The memory cell array by which the count information of rewriting which shows the count of rewriting of the block is stored in the redundancy field of the predetermined page of each block, Have a data register holding the data for 1 page, and data transfer between a data register and a memory cell array is performed per page. Elimination actuation builds in the flash EEPROM performed per block. It is RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment. The data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, A light demand is answered from said host equipment, and the light access means which carries out light access of said flash EEPROM is provided. This light access means A means to read the count information of rewriting from said predetermined page within the block for a light specified by said light demand, and to update the value of the count information of rewriting, Add said updated count information of rewriting to the light data which should be written in said predetermined page, and page data are generated. It is characterized by providing a means to transmit said page data to the data register of said flash EEPROM so that it may be stored in the data storage area and redundancy field said light data and said whose updated count information of rewriting are said predetermined pages.

[0025] In this RAM disk equipment, the count information of rewriting corresponding to that block is stored in the redundancy field of the predetermined page of each block, and the value of the count information of rewriting on that block for a light is updated in the case of the data writing to the block for a light. The updated count information of rewriting is added to the light data which should be written in a predetermined page, and is transmitted to the data register of a flash EEPROM. Since the count information of rewriting is managed within each block, it becomes unnecessary thus, to carry out the centralized control of the count information of rewriting in a specific field. For this reason, generating of the problem that the count of rewriting of a specific region projects and increases by light access for updating the count information of rewriting can be prevented. Moreover, only by light access to the same block, renewal of data writing and the count information of rewriting can carry out, and improvement in the light access engine performance can be aimed at as a result.

[0026] Moreover, invention concerning claim 8 includes two or more blocks which have two or more pages in which each includes a data storage area and a redundancy field, respectively. The memory cell array by which the count information of rewriting which shows the count of rewriting of the block is stored in the redundancy field of the predetermined page of each block, Have a data register holding the data for 1 page, and a command is answered from the outside. The elimination actuation which eliminates the content of storage per block, the light actuation which writes data in a memory cell array per page, And the flash EEPROM which performs lead actuation which reads data from a memory cell array in a page unit is built in. In the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment The data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, A light demand is answered from said host equipment, and the light access means which carries out light access of said flash EEPROM is provided. This light access means Publish a lead command and said flash EEPROM is set as a lead mode of operation. A means to read the count information of rewriting from said predetermined page within the block for a light specified by said light demand, and to update the value of the count information of rewriting, When the page of the outside for light access exists in the block for a light, The data stored in the page of the outside for [the] light access are read on said data buffer. The storing location on the data buffer of each data which constitutes the block data for 1 block which should be written in the block for [said] a light based on the address on the data buffer of the read data, and the address of the light data stored in said data buffer A means to generate the header table holding the shown address information, and a means to publish an erasion command, to set said flash EEPROM as an elimination mode of operation, and to eliminate the content of storage of said block for a light, With reference to the address information of said header table, the light data which should be written in said predetermined page are read from said data buffer. So that said updated count information of rewriting may be added to the light data, page data may be generated and it may be stored in the data storage area and redundancy field said light data and said whose updated count information of rewriting are said predetermined pages A means to transmit said page data to the data register of said flash EEPROM, With reference to the address information of said header table, the light data which should be written in in addition to said predetermined page are read from said data buffer one by one, and it is characterized by including a means to transmit it to the data register of said flash EEPROM.

[0027] In this RAM disk equipment, when the page of the outside for light access exists within the block for a light, reading appearance of the so-called involvement elimination data stored in the page of the outside for light access is carried out, and a

header table is generated based on the address of each of that involvement elimination data and light data from a host. If a header table is generated, after the content of storage of the block for a light is eliminated, according to the address information of a header table, block data will be transmitted to a data register in a page unit from a data buffer, and write-in actuation will be performed within a flash EEPROM. In this data transfer, the count information of rewriting is added and transmitted about the data which should be written in a predetermined page. For this reason, like the RAM disk equipment of claim 7, only by light access to the same block, renewal of data writing and the count information of rewriting can carry out, and improvement in the light access engine performance can be aimed at as a result. Moreover, data transfer processing to a flash EEPROM from a data buffer can be performed, without distinguishing involvement elimination data and light data by referring to this header table, since the header table holding the address information which shows the storing location of the block data for 1 block is generated.

[0028] Moreover, in addition to the configuration of claim 8, it is the configuration of having had error correcting code generation and the calculation function of claim 1, and, as for invention of claim 9, an error correcting code is stored in the redundancy field to which the count information of rewriting and an error correcting code are stored, and are each of other page in the redundancy field of the predetermined page of each block. In this RAM disk equipment, it becomes possible to use a redundancy field for validity more.

[0029] Moreover, invention of claim 10 carries out another **** management of the count information of rewriting on each block in the configuration of claim 7 at the high-order-bit section and lower bit section. It is the configuration of concentrating and storing the 1st count data of rewriting which consists of the high-order-bit sections in a predetermined block, and distributing and storing in the redundancy field of a predetermined page the 2nd count data of rewriting which consists of the lower bit sections for every block.

[0030] In this RAM disk equipment, since only the low order bit data of the count information of rewriting are stored in a redundancy field, the activity field of a redundancy field can be reduced. Moreover, whenever rewriting occurs, it is necessary to update low order bit data but, and since this updating is performed by light access for the data writing to the block for rewriting, and coincidence, it does not need to perform separately light access for data writing, and light access for renewal of the count information of rewriting like the RAM disk equipment of claim 7. Moreover, the count of rewriting of all blocks can be managed only by data read-out from one block, without reading the count information of rewriting sequentially from all blocks, since the data of the high-order-bit section of the count information of rewriting are concentrated and stored in the predetermined block.

[0031] Moreover, invention of claim 11 establishes the swapping means for equalizing the count of rewriting to the count management tool of rewriting of claim 10. In this RAM disk equipment, when a digit riser occurs to the data of the lower bit section of the count information of rewriting, it is examined whether the block exchange for leveling is performed. When the block with which the difference of the count of rewriting with the block for a light reaches beyond a predetermined value exists, the block is determined as a swapping point block, and the address and the content of storage are mutually replaced between the block and the block for a light. Thus, since it is examined whether the block exchange for leveling is performed on condition that the digit riser of the lower bit section data of the count information of rewriting, it becomes unnecessary to monitor the count of rewriting of each block continuously, and management of the count of rewriting becomes easy.

[0032] Moreover, invention of claim 12 stores a respectively discrete value for every block as initial value of the 2nd count data of rewriting in the time of disk manufacture and shipment etc. It can prevent the examination processing and swapping processing of swapping resulting from digit going up concentrating, and generating by this, even if rewriting to two or more blocks occurs continuously, since the stage when a digit riser occurs shifts for every block.

[0033] Moreover, in addition to the 1st count data of rewriting, the swapping table is stored in the predetermined block of FUSSHU EEPROM in invention of claim 13. The priority information which shows the priority of a swapping point block in this swapping table according to the value of the 1st count data of rewriting is held. Therefore, it becomes possible to specify a swapping point block easily, without carrying out sorting of the 1st count data of rewriting.

[0034]

[Example] Hereafter, the example of this invention is explained with reference to a drawing.

[0035] The configuration of the RAM disk equipment concerning one example of this invention is shown in drawing 1. This RAM disk equipment 10 is used as an alternative of a hard disk drive unit, connecting with host systems, such as a personal computer, emulates the disk-accessing demand from a host system (driver for a file system and magnetic disk drives) to the access request to a flash EEPROM, and accesses the flash EEPROM of RAM disk equipment 10 built-in. To this RAM disk equipment 10, it is possible to carry a maximum of 16 flash EEPROM chips. These flash EEPROMs are mounted in the circuit base of RAM disk equipment 10 free [attachment and detachment] through an IC socket, respectively. In drawing 1, the condition that 16 NAND mold flash EEPROMs 11-1 to 11-16 are carried is shown. A controller 12 and RAM13 are further formed in RAM disk equipment 10.

[0036] A flash EEPROM 11-1 to 11-16 is used as a record medium of this RAM disk equipment 10, and is equivalent to the magnetic-recording medium of a hard disk drive unit. These flash EEPROMs 11-1 to 11-16 are connected to the controller 12 through the NAND memory bus 14. The various control signal lines connected to the data bus of 8-bit width of face and the control signal terminal of each flash EEPROM are defined as the NAND memory bus 14. Common connection of an I/O data bus and the various control signal lines is made at data input/output terminal I/O of a flash EEPROM 11-1 to 11-16, and the control signal input terminal CONT (the command latch enabling terminal CLE, the address latch enabling terminal ALE, the write enable terminal WE, the lead enabling terminal RE, etc. are included), respectively.

[0037] Furthermore, between the flash EEPROM 11-1 to 11-6, and the controller 12, the chip select signal (CS1-CS8) line and the Ready/Busy signal line (R/B1 - R/B16) are independently arranged for every chip, respectively.

[0038] These flash EEPROMs 11-1 to 11-6 are the 16M bit NAND molds EEPROM, respectively, and they are equipped with the memory cell array 111 and the data register 112 as shown in drawing 2. The memory cell array 111 has a 8K line x264 train x8 bit bit pattern, and has 512 blocks. Data elimination can be performed in this block unit. Each block consists of 16 pages (line), and each page is equipped with 256 bytes of data storage area, and 8 bytes of redundancy field. Writing and read-out of data are performed per page through 256+8 bytes of data register 112. The redundancy field of each page is used for storing of ECC or the count data of rewriting in this RAM disk equipment.

[0039] The flash EEPROM 11-1 to 11-6 has modes of operation, such as a write mode, a Read mode, and washout mode, respectively, and these modes of operation are specified by the command from a controller 12.

[0040] In a Read mode, automatic activation of the data transfer (page lead) of the page unit from the memory cell array 111 to a data register 112 is carried out inside a flash EEPROM. a Ready/Busy signal line is set as a Busy condition during this data transfer period -- having -- this -- a page lead -- a working thing is notified to a controller 12. Reading appearance of the data for 1 page transmitted to the data register 112 is serially carried out by 8 bitwises from input/output terminal I/O0 - I/O7. Termination of the serial lead of the data for 1 page performs the page lead of the following page automatically.

[0041] In a write mode, automatic activation of the data transfer (page light) of the page unit from the data register 112 to the memory cell array 111 is carried out inside a flash EEPROM. a Ready/Busy signal line is set as a Busy condition during this data transfer period -- having -- this -- a page light -- a working thing is notified to a controller 12.

[0042] In washout mode, package elimination of the content of storage of a block of the arbitration specified by the command from a controller is carried out.

[0043] In RAM disk equipment 10, most processings for control of these flash EEPROMs 11-1 to 11-6 are controlled by firmware. The ECC check in the case of the ECC generation in the case of the data writing to a flash EEPROM and data read-out from a flash EEPROM is performed by hardware.

[0044] Next, the configuration of the disk controller 12 of RAM disk equipment 10 is explained.

[0045] The disk controller 12 is realized by one LSI, and accumulation formation of a microprocessor (MPU) 14, the processor bus interface 15, ROM16, the external bus interface 17, the buffer memory controller 18, and the NAND bus interface 19 is carried out at the LSI chip as shown in drawing 1.

[0046] A microprocessor 14 is what performs firmware stored in ROM16 and controls actuation of this RAM disk equipment 10 whole. The host address for disk accessing sent from CPU1 of a personal computer The address translation processing changed into the memory address for accessing a flash EEPROM 11-1 to 11-16, the access-control processing for carrying out read/write access of the flash EEPROM 11-1 to 11-16, and a flash EEPROM 11-1 to 11-6 -- control of the swapping processing for equalizing management of each count of data rewriting and the count of data rewriting etc. is performed.

[0047] Various programs, such as firmware performed by the microprocessor 14 and an operating system performed by CPU1 of a personal computer, and an application program, are stored in ROM16. the operating system and the application program are managed as a file within ROM16, and these programs are published from CPU1 to RAM disk equipment 10 -- reading appearance is carried out, and reading appearance is carried out from ROM16 according to a demand, and it is loaded to the main memory 3 of a personal computer.

[0048] The continuous disk address is mapped by ROM16 and the flash EEPROM 11-1 to 11-16, and ROM16 is also used for them as some storages of a disk unit 10. For this reason, read-out of the operating system and application program which are stored in ROM16 is performed only when the read-out demand from CPU1 specifies a certain specific address, and when specifying other addresses, read access to a flash EEPROM 11-1 to 11-16 is performed.

[0049] The external bus interface 17 is an interface based on IDE or a PCMCIA specification, and is connected to the system bus 2 of a personal computer. The I/O register group for the communication link with CPU1 is prepared in the external bus interface 17. The sector number register 171, the sector count register 172, a data register 173, the cylinder register 174, the drive / head register 175, the command register 176, the status register 177, etc. are contained in the register group.

Read/write is possible for these registers respectively by the microprocessor 14 and CPU1 of a personal computer.

[0050] The light of the sector number which specifies an access head location is carried out to the sector number register 171 by CPU1. The light of the number of sectors for read/write is carried out to the sector count register 172 by CPU1. The lead data by which reading appearance is carried out from the light data or the RAM disk equipment 10 supplied from CPU1 are set to a data register 173. The light of the cylinder number for read/write is carried out to the cylinder register 174 by CPU1. The light of the drive number for read/write and the head number is carried out to a drive / head register 175 by CPU1. The light of a lead command, a light command, etc. which specify actuation of RAM disk equipment 10 is carried out to a command register 176 by CPU1. The various statuses of the RAM disk equipment 10 which should be notified to CPU1 are set to a status register 177.

[0051] The buffer memory controller 18 is for carrying out the access control of RAM13, and performs data transfer between a system bus 2 and RAM13 etc. There is data transfer between others, RAM13, flash EEPROM 11-1 - 11 -16 [data transfer / between a system bus 2 and RAM13] among the data transfer. The latter data transfer is performed by activation of the MUBU string instruction by the microprocessor 14.

[0052] The NAND bus interface 19 is hardware logic which carries out the access control of the flash EEPROM 11-1 to 11-16 to the bottom of control of a microprocessor 14, and delivers and receives the data transfer between flash EEPROMs 11-1 to 11-16, and the various control signals between flash EEPROMs. The I/O register group 20 in which read/write is possible,

and the ECC arithmetic circuit 21 are established in this NAND bus interface 19 by the microprocessor 14. The ECC arithmetic circuit 21 performs the operation for the operation for ECC generation, and an ECC check.

[0053] In the I/O register group 20, as shown in drawing 3 The NAND chip number register 201, the NAND data read/write register 202, the NAND chip enable control register 203, the NAND command latch enabling control register 204, the NAND address latch enabling control register 205, The ECC start control register 206, ECC data JISUTA 209-214 of an ECC ready / busy status register 207, and 208 or 6 LWC registers, the 16 NAND chip readies / busy status registers 215-1 to 215-16, etc. are contained.

[0054] The NAND chip number register 201 is a register for choosing to which NAND mold flash EEPROM processing is performed, and the value which shows the chip number of the NAND mold flash EEPROM for access is set by the processor 14.

[0055] The NAND data read/write register 202 is a register for performing read/write of the data to a flash EEPROM through a data bus, and is used for issuance of the command to a flash EEPROM, issuance of the address, the lead of status data, etc.

[0056] The NAND chip enable control register 203 is a register for controlling the chip enable terminal CE of a flash EEPROM, and the flag which shows generating / un-generating is set by the processor 14. [of a chip select signal] If a flag "1" is set to a bit 0, the NAND bus interface 19 will generate the chip select signal (CS) for activating the chip enable terminal CE of the flash EEPROM chosen with the NAND chip number register 201.

[0057] The NAND command latch enabling control register 204 is a register for controlling the command latch enabling terminal CLE of a flash EEPROM, and the flag which specifies whether the command latch enable terminal CLE is activated is set by the processor 14. If a flag "1" is set to a bit 0, the command latch enabling terminal CLE of a flash EEPROM will be actively set up with the NAND bus interface 19. Thereby, it is notified to a data bus to a flash EEPROM that the command was published.

[0058] The NAND address latch enabling control register 205 is a register for controlling the address latch enabling terminal ALE of a flash EEPROM, and the flag which specifies whether the address latch enable terminal ALE is made into the bitter taste eve is set by the processor 14. If a flag "1" is set to a bit 0, the address latch enabling terminal ALE of a flash EEPROM will be actively set up with the NAND bus interface 19. Thereby, it is notified to a data bus to a flash EEPROM that the address was published.

[0059] The ECC start control register 206 is a register for controlling activation initiation of an ECC operation, and before starting the data transfer between RAM13 and a flash EEPROM 11-1 to 11-16, the flag which specifies activation initiation of an ECC operation is set by the processor 14. "1" is set to a bit 0 when performing data transfer from a flash EEPROM to RAM13. In this case, the ECC arithmetic circuit 21 performs an ECC check about the data by which reading appearance is carried out from a flash EEPROM. "0" is set to a bit 0 when performing data transfer from RAM13 to a flash EEPROM. In this case, the ECC arithmetic circuit 21 generates ECC about the data transmitted from RAM13.

[0060] An ECC ready / busy status register 207 is status registers which the ECC operation (ECC generation or ECC check) by the ECC arithmetic circuit 21 shows whether it is under [activation] *****. The NAND bus interface 19 sets the flag of "0" to a bit 0, after setting the flag of "1" to a bit 0 and completing activation of an ECC operation, if it is among ECC instruction execution. If a processor 14 investigates the existence of termination of ECC data processing with reference to the flag of a register 207 after 256 bytes of data transfer termination between RAM13 and a flash EEPROM 11-1 to 11-16 and termination is checked, it will start 256 bytes of next data transfer processing.

[0061] The data which should be stored in 2 bytes of the head of 8 bytes of redundancy field of a flash EEPROM, for example, the LWC (Low Write Count) data which are the lower bit of the count data of rewriting, are set to the LWC register 208. In this case, the LWC data corresponding to the block for a light of a flash EEPROM which serves as the destination from RAM13 in advance of that data transfer in the data transfer to a flash EEPROM are set by firmware. In the data transfer from a flash EEPROM to RAM13, LWC data are taken out from the data stream by which reading appearance is carried out from a flash EEPROM with the NAND bus interface 19, and it is set.

[0062] The ECC data registers 209-214 are the register groups for holding 6 bytes of ECC data. At the time of the data transfer from RAM13 to a flash EEPROM, 6 bytes of ECC data corresponding to 256 bytes of light data are generated by the ECC arithmetic circuit 21, and the generated ECC data is set to the ECC data registers 209-214. At the time of the data transfer from a flash EEPROM to RAM13, the result of the ECC check operation performed by the ECC arithmetic circuit 21 is set to registers 209-214. It performs, in order that this ECC check operation may perform error detection. In this ECC check operation, 6 bytes of ECC data corresponding to 256 bytes of lead data are generated again, and the comparison with 6 bytes of ECC data by which reading appearance was actually carried out to 6 bytes of that ECC data from the flash EEPROM is performed per cutting tool. It will be set to "0" and the bit 0 of the register with which "1" corresponds if inharmonious if in agreement. Therefore, if the error has not occurred, all the values set to the bit 0 of the ECC data registers 209-214 are set to "0."

[0063] a NAND chip ready / busy status register 215-1 to 215-16 -- 16 flash EEPROMs 11-1 to 11-16 -- it is a register group holding each ready/busy status, and the flag which shows whether a corresponding flash EEPROM is a ready state or it is a busy condition is set to the bit 0 of each register with the NAND interface 19. If it is a ready state, as for that, a flag "0" will be set for a flag "1" by the busy condition. It is judged by the ready/busy signal (R/B1 - R/B16) independently inputted from a flash EEPROM 11-1 to 11-16, respectively for every chip whether it is a ready state or it is a busy condition.

[0064] RAM13 of drawing 1 consists of dynamic RAMs, and a part of the storage region is used as a data buffer 131 holding the light data transmitted from CPU1 of a personal computer, and the lead data by which reading appearance is carried out

from a flash EEPROM 11-1 to 11-16. Moreover, RAM13 is used as a working area of a microprocessor 14, and the count of rewriting managed table 134 for managing the address translation table 132 for changing the host address into a memory address, the header table 133 which manages the storing location of the block data for 1 block which should be written in a flash EEPROM, and the count of rewriting of a flash EEPROM 11-1 to 11-16 per block etc. is stored here.

[0065] Next, with reference to drawing 4, the memory map in the RAM disk equipment 10 managed by the microprocessor 14 is explained.

[0066] The RAM field, the NAND memory data area, the application ROM (APL-ROM) field, and the firmware ROM (FW-ROM) field are mapped like a graphic display by the memory address space which a microprocessor 14 manages.

[0067] A RAM field is the memory address space for accessing RAM13, and the data buffer 131 etc. is assigned here.

[0068] A NAND memory data area is 256 bytes of address space for accessing a flash EEPROM 11-1 to 11-16. 256 bytes of data transfer is performed by the MUBU string instruction of a microprocessor 14 in read/write access of the data to a flash EEPROM 11-1 to 11-16. For this reason, 256 bytes of address space is secured as a NAND memory data area for 256 bytes of this data transfer. By using this NAND memory data area, all data transfer between a data buffer 131 and a flash EEPROM 11-1 to 11-16 can be performed as processing of firmware.

[0069] next, drawing 5 -- referring to -- a flash EEPROM 11-1 to 11-16 -- the utilization gestalt of each block is explained.

[0070] The block of a flash EEPROM 11-1 to 11-16 is used not only for storing of user data but for storing of the various management information for controlling these flash EEPROMs. The assignment of the user data and management information to a flash EEPROM 11-1 to 11-16 is as follows.

[0071] That is, in the NAND mold flash EEPROM 11-1 of chip number #0, 504 blocks of block 0 - block 503 are used as memory block for storing user data in 512 blocks like a graphic display. Moreover, block 504 is used as a MBI (MemoryBlock Information) table block which memorizes the management information for managing the address, a count of rewriting, etc. of 512 blocks in a chip. Moreover, blocks 505-509 are used as a preliminary block for the alternative, when a defect occurs in memory block and a MBI table block. Block 511 is used as a KONFIGU block which memorizes the configuration information of RAM disk equipment 10, and block 510 is used as a preliminary block for an alternative of a KONFIGU block.

[0072] Only one KONFIGU block exists by the equipment 10 whole, and the information which shows the number of chips of the flash EEPROM carried in RAM disk equipment 10, the MBI table block in each flash EEPROM existence location, etc. is managed here.

[0073] In the NAND mold flash EEPROM 11-2 to 11-16 of chip number #1-#16, like a graphic display, 504 blocks of block 0 - block 503 are used as memory block for storing user data, and block 504 is used as a MBI (Memory Block Information) table block which memorizes the management information for managing the address and the count of rewriting of 512 blocks in a chip. Moreover, blocks 505-511 are used as a block for an alternative of memory block and a MBI table block.

[0074] Although the physical block structure of memory block, a MBI table block, an alternative block, and each KONFIGU block is as drawing 2 having explained, the data storage format within these blocks is explained hereafter.

[0075] The data storage format of memory block for storing user data is shown in drawing 6.

[0076] As drawing 2 explained, the NAND mold flash EEPROM used with this RAM disk equipment 10 is a 16M bit chip, and each block consists of 16 pages from a page 0 to a page 15. Each page consists of 256 bytes of a data area, and 8 bytes of a redundancy field.

[0077] Generally, disk accessing by the host system is performed per 512 bytes of sector. In this case, the size of 256 bytes of user data stored in the data area of each page serves as half [of the user data size for 1 sector]. In this RAM disk equipment 10, only 256 bytes of data area is used for storing of user data, and the redundancy field which is 8 bytes is not used for storing of user data. Therefore, 512 bytes of user data are continued and stored in 2 pages followed within a block in this RAM disk equipment 10. For this reason, the user data stored in 1 block turn into the data for 8 sector, for example, the data from a sector 0 to a sector 7.

[0078] That is, to 256 bytes in the first half of a sector 0, and a page 1, 256 bytes in the second half of a sector 1 are stored in a page 0 in 256 bytes in the first half of a sector 1, and a page 3 at 256 bytes in the second half of a sector 0, and a page 2, and user data are stored per 1 / 2 sectors like the following as shown in drawing 5.

[0079] pages 0-15 -- 8 bytes of each redundancy field is divided into 2 bytes of field, and 6 bytes of field, and ECC is stored in 6 bytes of the field. This ECC corresponds to 256 bytes of user data stored in the data area of the same page.

[0080] Namely, ECC stored in the redundancy field of a page 1 is similarly generated from 256 bytes of user data stored in the data area of a page 1 by generating ECC stored in the redundancy field of a page 0 from 256 bytes of user data stored in the data area of a page 0.

[0081] Thus, in this RAM disk equipment 10, ECC is generated not per 512 bytes of sector unit but per 256 bytes of page, and the data stream and ECC for an operation are stored in the same page. For this reason, even if the error of a flash EEPROM that a two or more pages defect cel occurs in each same bit position occurs like a graphic display, the defect cel contained in the data stream set as the operation object of ECC will be set only to one. Therefore, it becomes possible to perform error detection and correction by the easy usual ECC operation, without using the complicated high ECC operation expression of data bounce-back capability.

[0082] Moreover, LWC data are stored in the remaining 2 bytes of the redundancy field of a page 0 as block status information of block 0. This LWC data consists of data of the 10 bits of lower bit sections of the count data of rewriting in which the count of rewriting of block 0 is shown, for example, the low order of the count data of rewriting of 26-bit width of

face.

[0083] one LWC data is stored for every block -- having -- block 1 - block 503 -- LWC data are similarly stored in 2 bytes of the head of the redundancy field of each head page as the block status information.

[0084] Furthermore, BAD sector information and ECC error information are stored in the remaining 2 bytes of the redundancy field of a page 1 as sector status information of a sector 0.

[0085] BAD sector information is status information the sector 0 indicates it to be whether it is a bad sector. the same -- the three-page page 5 and a page 15 -- the remaining 2 bytes of each redundancy field -- the 1 sector sector 2 and .. a sector 7 - it is alike, respectively and the BAD sector information that it corresponds is stored. These BAD(s) sector information is given by CPU1 of the personal computer, when the personal computer which is the host system of RAM disk equipment 10 executes the formatting command for formatting RAM disk equipment 10.

[0086] ECC error information is status information which shows the existence of ECC error generating to the sector. For this ECC error information, a personal computer is WRITE. When the data writing by the LONG command is performed and the error of the burst error used as an ECC error etc. occurs, it is given by CPU1 of a personal computer.

[0087] Next, with reference to drawing 7, the data storage format of a MBI table block is explained.

[0088] The MBI table block consists of one block containing 16 pages from a page 0 to a page 15. Each page consists of 256 bytes of a data area, and 8 bytes of a redundancy field. ECC corresponding to 256 bytes of data stored in the data area of the page is stored in 6 bytes of the back of 8 bytes of the redundancy fields. Moreover, 2 bytes of the head of a redundancy field are intact.

[0089] The address translation table, the UWC table, and the swap order table are stored in the data area of a MBI table block like a graphic display.

[0090] The address translation table in a MBI table block holds the address translation information for changing the physical block addresses 0-511 of the blocks 0-511 in a chip into the host address to which it is specified with a cylinder number, a head number, and a sector number. The host address is stored in order of the physical block address.

[0091] Here, as a value of the host address in a MBI table block, the host logical address calculated from the cylinder number given from a host system, a head number, and not the sector number itself but these cylinder numbers, the head number, and the sector number is used.

[0092] This host logical address is the serial number of the sector at the time of assigning the sector number which continues sequentially from "0" to all the sectors (= 8x512x16 sectors) prepared in a flash EEPROM 11-1 to 11-16. Therefore, the bit width of face of the host logical address becomes 16 bits. However, there is no address translation information managed by the address translation table of this MBI table block per 1 sector, and since it is for performing address translation per 8 sectors which are the number of sectors contained in 1 block, only 13 bits of high orders in the 16-bit host logical address are actually stored on the table.

[0093] A UWC table holds the UWC (UPPER WRITECOUNT) data which consist of the high-order-bit sections of the count data of rewriting in which the count of rewriting of each 512 blocks in a chip is shown. UWC data have 16-bit width of face, and if this UWC data and LWC of the above-mentioned 10-bit width of face are doubled, they can count it to rewriting of 64 million counts at the maximum. The value of UWC data is counted up +one time, whenever a digit riser occurs to corresponding LWC data (i.e., whenever 1024 rewritings occur).

[0094] These address translation tables and a UWC table are stored in the data area of a page 5 to the page 12 of a MBI table block like a graphic display. An address translation table and a UWC table are transmitted to RAM13 at the time of equipment starting, i.e., the power-on of RAM disk equipment 10, and reside permanently as an address translation table 132 and count of rewriting managed tables 134, respectively.

[0095] A swap order table is a table referred to by the microprocessor 14 in the case of the swapping actuation for block exchange, and holds the block address which shows the block (swapped block) of the swapping point in order of a priority (order with the small value of UWC data). The swap order table is stored in the data area of a page 13 to the page 15 of a MBI table block like a graphic display. A swap order table does not reside in RAM13 permanently, but at the time of examination of block exchange, i.e., digit riser generating to UWC data from LWC data, it is transmitted to RAM13 and refer to it for a microprocessor 14.

[0096] The configuration of an address translation table and a UWC table is shown in drawing 8.

[0097] Like a graphic display, the host address and the storing field of UWC data have the data size of 2 bytes, respectively. Here, since the host address is 13 bits as mentioned above, the high order triplet (a bit 15 - bit 13) of 2 bytes of storing field is not used. The host address and UWC data are arranged by turns in order of the physical block address.

[0098] That is, the host address and UWC data are stored in order by 64 blocks (block 0 - block 63) from the head in 256 bytes of data area of the page 5 in a MBI table block in the sequence of the UWC data corresponding to the UWC data corresponding to the host address corresponding to the physical block address 0, and the physical block address 0, the host address corresponding to the physical block address 1, and the physical block address 1.

[0099] Similarly, the host address and UWC data are stored in order by 64 blocks (block 448 - block 511) from the head in 256 bytes of data area of a page 12 in the sequence of the UWC data corresponding to the UWC data corresponding to the host address corresponding to the physical block address 448, and the physical block address 448, the host address corresponding to the physical block address 449, and the physical block address 449.

[0100] The configuration of a swap order table is shown in drawing 9.

[0101] The swap order table is divided into the 1st order table corresponding to 256 blocks of the first half from block 0 to

block 255, and the 2nd order table corresponding to 256 blocks of the second half from block 256 to block 511.

[0102] It sets on the 1st order table and the value of the minimum UWC in block 255 is stored in 2 bytes of field (addresses 800h and 801h) of the head from the block 0.

[0103] the following 1 byte (address 802h) -- a minimum of -- the value which shows the total of the block with the value of UWC is stored.

[0104] Furthermore, the value which shows the total of the block with the value of minimum UWC+1 is stored in the following 1 byte (address 803h). here -- "minimum UWC+1" -- to the last -- a minimum of -- the value which added 1 to UWC -- being shown -- **** -- a minimum of -- small UWC is not shown in the degree of UWC. A block total with the value of minimum UWC+1 may serve as zero.

[0105] Henceforth, sequential storing of the data in which a block total with the value of minimum UWC+120 is shown is carried out from the data in which a block total with the value of minimum UWC+2 is shown at address 804h-87Ah.

[0106] The data in which the count of rewriting of the MBI table block [itself] is shown are stored in address 87Ch-87Fh.

[0107] The information as an original swap order table is stored in address 880h-97Fh. namely, -- as the value which shows the block for [of whenever / of the highest priority / swapped] to address 880h in block 0 - block 255 -- a minimum of -- the physical block address of a block which has the value of UWC is stored. When two or more blocks which have the value of the minimum UWC exist, priority becomes [a physical block address] high from the rearmost block at order. Hereafter, the block address of the block for a swap is similarly stored in 97Fh(s) in order of a priority from address 881h.

[0108] Next, the 2nd order table is explained.

[0109] the 2nd order table -- it is and the value of the minimum UWC in block 511 is stored in 2 bytes of field (addresses 980h and 981h) of the head from the block 256 except for the MBI table block, the alternative block, and the KONHIIGU block. That is, since, as for an alternative block etc., the count of rewriting is not advancing at all, the count of rewriting of these blocks is disregarded. Moreover, when an alternative occurs actually, suppose the value of the UWC data of the alternative block that UWC of a defective block is inherited not the actual count of rewriting of the alternative block but a substituting agency that is,. This is for preventing the degradation by frequent occurrence of swapping actuation. If the value of the actual UWC data of an alternative block is adopted as it is, the value of the count of rewriting projects, a small block will appear suddenly and the swapping actuation which sets the alternative block as the object-ed [swapping] will be generated repeatedly.

[0110] the following 1 byte (address 982h) -- a minimum of -- the value which shows the total of the block with the value of UWC is stored.

[0111] Furthermore, the value which shows the total of the block with the value of minimum UWC+1 is stored in the following 1 byte (address 983h). Henceforth, sequential storing of the data in which a block total with the value of minimum UWC+120 is shown is carried out from the data in which a block total with the value of minimum UWC+2 is shown at address 984h-9FAh.

[0112] The information as an original swap order table is stored in address A00 h-AFFh. namely, -- as the value which shows the block for [of whenever / of the highest priority / swapped] to address A00h in block 256 - block 511 -- a minimum of -- the physical block address of a block which has the value of UWC is stored. When two or more blocks which have the value of the minimum UWC exist, priority becomes [a physical block address] high from the rearmost block at order. Hereafter, the block address of the block for a swap is similarly stored in AFFh in order of a priority from address A01h.

[0113] Thus, in the swap order table, the block used as the swapping point is stored in order of the priority. For this reason, it becomes possible to specify the swapping point, without a microprocessor 14 performing sorting of UWC data by referring to this swap order table.

[0114] The data storage format of a KONHIIGU block is shown in drawing 10.

[0115] The KONHIIGU block consists of one block containing 16 pages from a page 0 to a page 15. Each page consists of 256 bytes of a data area, and 8 bytes of a redundancy field. ECC corresponding to 256 bytes of data stored in the data area of the page is stored in 6 bytes of 8 bytes of redundancy field. Moreover, the remaining 2 bytes of a redundancy field are intact.

[0116] KONHIIGU information, CIS information, and drive ID information are stored in the data area of a KONHIIGU block like a graphic display.

[0117] KONHIIGU information defines the physical block address which shows the number of chips of the flash EEPROM carried in this RAM disk equipment 10, the number of the maximum chips which can be written in simultaneous, and the MBI table block in each chip existence location. This KONHIIGU information is referred to by the microprocessor 14 at the time of the power-on of RAM disk equipment 10, and the initialization process according to that KONHIIGU information is performed.

[0118] CIS information is card attribute information notified to a host system, when using RAM disk equipment 10 as a card of a PCMCIA specification. The information about physical specifications, such as a class of memory chip, capacity, and access time, and the information about the application which shows that this card is a disk unit etc. are defined as this card attribute information. This CIS information is transmitted to RAM13 by the microprocessor 14 at the time of the power-on of RAM disk equipment 10. The CIS information transmitted to RAM13 answers the read-out demand of the predetermined address from a host system, and is transmitted to a host system.

[0119] Drive ID information is information which shows the drive configuration of RAM disk equipment 10, and the total number of sectors logically assigned to a built-in flash EEPROM, the number of cylinders, the number of heads, the number of sectors per one track, etc. are defined here. This drive ID information answers a drive discernment command from a host

system, and is transmitted to a host system by the microprocessor 14.

[0120] Thus, in this example, ECC corresponding to the data stored in the data area of that page is stored in the redundancy field of each page also in the MBI table block and KONHIIGU block which are used not only for memory block for storing user data but for management information storing. Therefore, also in these MBI(s) table block and a KONHIIGU block, it becomes possible to perform error detection and correction by the easy usual ECC operation, without using the complicated high ECC operation expression of data bounce-back capability.

[0121] Next, with reference to drawing 11, the configuration of the address translation table 132 prepared in RAM13 is explained.

[0122] a thing for this address translation table 132 to change the host address specified with a cylinder number, a head number, and a sector number into the physical memory address for accessing a flash EEPROM 11-1 to 11-116 -- it is -- this address translation table 132 -- a flash EEPROM 11-1 to 11-116 -- it is generated based on the address translation information on each MBI table block.

[0123] In this address translation table 132, a 13-bit block sector address is used as the host address. This block sector address consists of 13 bits of high orders in the host logical address (16 bits) which shows the serial number of the sector calculated from the cylinder number, the head number, and the sector number, as mentioned above. Moreover, a physical memory address consists of the 4-bit chip number address and a physics block address in [of 9 bits] a chip. The 4-bit chip number address specifies one of 16 chips from the flash EEPROM 11-1 of chip number #0 to the flash EEPROM 11-6 of chip number #15. The physics block address in [of 9 bits] a chip specifies one in 512 blocks of the block 0 in the chip specified by the chip number address - block 511. The physical memory address (the chip number address, physics block address in a chip) is stored in order of the host address.

[0124] Next, with reference to drawing 12, the principle of the address translation actuation using an address translation table 132 is explained.

[0125] First, the cylinder number, head number, and sector number which were specified from the host system are changed into the host logical address of 16-bit width of face by the firmware performed by the microprocessor 14. Subsequently, an address translation table 132 is searched by making into an index the block sector address which consists of 13 bits of high orders of the host logical address of the 16-bit width of face, and the chip number address of the 4-bit width of face corresponding to a block sector address and the physical block address of 9-bit width of face are determined.

[0126] The chip number address is set to the NAND chip number register 201 of the NAND bus interface 19 by firmware for a chip select. Moreover, the sector address within a block which consists of the low order triplet of the host logical address of 16-bit width of face, the even-/recto appointed address of 1-bit width of face, and the column address of 8-bit width of face are added to a 9-bit physical block address within NAND BAIINTAFESU 19, and the memory address (A20-A0) of 20-bit width of face is generated. This memory address (A20-A0) is set to the NAND data read/write register 202 of the NAND bus interface 19 for addressing. Since access of the RAM disk equipment 10 performed by the host system is performed per sector in order to carry out read/write of the user data, both of even-/recto appointed addresses (A8) of 1-bit width of face and column addresses (A7-A0) of 8-bit width of face are usually oar "0."

[0127] Next, with reference to drawing 13, ECC generation / check actuation of the ECC arithmetic circuit 21 of the NAND bus interface 19 is explained.

[0128] First, generation actuation of ECC is explained with reference to drawing 13 (a).

[0129] Reading appearance of the user data stored in the data buffer 131 of RAM13 at the time of the data writing to the NAND mold flash EEPROM 11-1 to 11-16 is carried out by the microprocessor 14 per 256 bytes, and they are transmitted to the data register of a flash EEPROM through the NAND bus interface 19.

[0130] In the NAND bus interface 19, 256 bytes of user data are sent to the ECC arithmetic circuit 21, and the operation for generating ECC there is performed. Here, Hamming code etc. is used as ECC.

[0131] An ECC operation is performed 256 bytes of not only user data but for 258 bytes of data which consist of the 256 bytes of user data and data which should be stored in the first 2 bytes of a redundancy field actually.

[0132] For example, in the data writing to the head page of each block, 258 bytes of data which consist of 256 bytes of user data and 2 bytes of LWC data are set as the object of an ECC operation. 2 bytes of this LWC data is set to the LWC register 208 by the microprocessor 14 in advance of 256 bytes of user data transfer.

[0133] The ECC arithmetic circuit 21 calculates 258 bytes of data which consist of 256 bytes of user data, and 2 bytes of LWC data set to the LWC register 208 per predetermined data stream, and carries out sequential generation of 6 bytes of Hamming code ECC. These Hamming codes ECC are held temporarily at the ECC data registers 209-214.

[0134] If generation processing of ECC is completed, the ECC arithmetic circuit 21 will be made to follow 256 bytes of user data, and will transmit 2 bytes of LWC data, and 6 bytes of ECC data to the data register of a flash EEPROM. Thereby, the data for 1 page which consists of 256 bytes of user data, 2 bytes of LWC data, and 6 bytes of ECC data are set to the data register of a flash EEPROM.

[0135] Next, check actuation of ECC is explained with reference to drawing 13 (b).

[0136] The data for 1 page by which reading appearance is carried out from a flash EEPROM at the time of data read-out from the NAND mold flash EEPROM 11-1 to 11-16 are sent to the NAND bus interface 19.

[0137] In the NAND bus interface 19, 256 bytes of user data are picked out from the data for 1 page, and it is sent to the data buffer 131 of RAM13. Moreover, the data for 1 page is sent to the ECC arithmetic circuit 21.

[0138] First, the ECC arithmetic circuit 21 picks out LWC data from the data for 1 page, sets it to the LWC register 208, and

performs the ECC check operation for error detection after that.

[0139] In this ECC check operation, the ECC arithmetic circuit 21 calculates 258 bytes of data which consist of 256 bytes of user data, and 2 bytes of LWC data in the same data stream unit as the case of ECC generation processing, and carries out sequential generation of 6 bytes of Hamming code ECC. And Hamming code ECC contained in the data for 1 page by which reading appearance was carried out from the generated Hamming code ECC and flash EEPROM is compared per cutting tool, and the comparison result which shows coincidence/inequality is set to the ECC data registers 209-215. When there is no error generating, the data "0" in which coincidence is shown are altogether set to the ECC data registers 209-215. If the content of the ECC data registers 209-215 was referred to by the microprocessor 14 and the error has occurred, the error correction of the data on a data buffer 131 will be performed by the microprocessor 14.

[0140] Next, in order to make clear the functional assignment of the firmware performed by micro PUSESSA 14 and the NAND bus interface 19, the access-control actuation to a flash EEPROM is explained.

[0141] First, with reference to drawing 14 and drawing 15, processing of firmware required for control of the program cycle of a flash EEPROM and actuation of the NAND bus interface 19 are explained.

[0142] Drawing 14 is a flow chart explaining the processing of firmware and the actuation of the NAND bus interface 19 in a program cycle, and drawing 15 is a timing chart which shows actuation of the flash EEPROM in a program cycle.

[0143] Here, 256 bytes of user data, 2 bytes of LWC data, and 6 bytes of ECC data are explained to the head page of a certain block taking the case of the case where a light is carried out.

[0144] With reference to one of the NAND chip ready / busy registers 215-1 to 215-16 first corresponding to the flash EEPROM chip for a light in the firmware performed by micro PUROSSE 14, the flash EEPROM chip for [the] a light checks whether you are the Lady State (waiting) (step A11). If it checks that he is the Lady State, firmware will carry out the I/O light of the NAND chip number register 201, and will set the chip number corresponding to the flash EEPROM chip for a light there (step A12).

[0145] Subsequently, firmware carries out the I/O light of the NAND chip enable control register 203, and sets CE on-command which specifies activating a chip enable input (CE) there (step A13).

[0146] At this time, the chip select signals CS1-CS16 corresponding to the chip number by which the NAND bus interface 19 was set to the NAND chip number register 201 activate one (step B11). This chip select signal is supplied to chip enable input CE~ of the chip for a light, and the chip for a light is set as enabling state by this.

[0147] Next, firmware carries out the I/O light of the NAND command latch enabling control register 204, and sets the CLE on-command which specifies activating a command latch enabling (CLE) signal there (step A14).

[0148] The NAND bus interface 19 answers the set of a CLE on-command, and sets the command latch enable signal CLE to an active state, i.e., "H" level, (step B12).

[0149] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets a serial data input command "80h" there (step A15). The NAND bus interface 19 answers the set of a serial data input command "80h", and sets write enable signal WE~ to an active state, i.e., "L" level, and transmits a serial data input command "80h" to input/output terminal I/O (step B13).

[0150] Firmware carries out the I/O light of the NAND address latch enabling control register 205, and sets an ALE on-command there (step A16). The NAND bus interface 19 answers the set of an ALE on-command, and sets address latch enable signal ALE to an active state, i.e., "H" level, (step B14).

[0151] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets a memory address (column address A7-A0 and page address A20-A8) there (step A17). The NAND bus interface 19 answers the set of the address, write enable signal WE~ is set to an active state, i.e., "L" level, and the set memory address is transmitted to input/output terminal I/O (step B15).

[0152] Firmware sets to the LWC register 208 the LWC data which read from the block for a light beforehand and were counted up +one time before this program cycle activation (step A18). And firmware sets an ECC initiation command to the ECC start control register 206 (step A19), and transmits 256 bytes of user data to a flash EEPROM from a data buffer 131 by activation of a MUBU string instruction after that (step A20).

[0153] The NAND bus interface 19 answers an ECC initiation command, starts the operation for ECC generation, and generates 6 bytes of ECC data corresponding to 256 bytes of user data, and LWC data (step B16). And the NAND bus interface 19 follows 256 bytes of user data, and starts 2 bytes of LWC data, and 6 bytes of ECC data transfer (step B17). Thereby, 256 bytes of user data, 2 bytes of LWC data, and 6 bytes of ECC data are transmitted as serial input data (DIN) for 1 page as shown in the data register of the flash EEPROM for a light at drawing 15.

[0154] Then, after firmware publishes a CLE on-command, it carries out the I/O light of the NAND data read/write register 202, and sets a program command "10h" there (step A21). First, the NAND bus interface 19 sets write enable signal WE~ as an active state, after setting the command latch enable signal CLE as an active state, and it transmits a program command "10h" to input/output terminal I/O (step B18).

[0155] A transfer of this program command "10h" carries out automatic activation of page light actuation and the verification actuation in the flash EEPROM for a light.

[0156] In page light actuation, page data transfer for 1 page from a data register to a memory cell array is performed, and 256 bytes of user data of a data register are written in the data area of the page, and 2 bytes of LWC data of a data register and 6 bytes of ECC data are written in the redundancy field of the page. During this page write-in actuation period, the ready/busy signal from that chip (RDY/BSY) will return from a busy condition to a ready state, if a busy condition is shown and write-in

actuation is completed.

[0157] If completion of write-in actuation is checked according to the content of the NAND chip ready / the busy register corresponding to the chip which carried out light access (step A22), firmware will carry out the I/O light of the NAND data read/write register 202, and will set a status lead command "70h" there (step A23). The NAND bus interface 19 answers the set of a status lead command "70h", sets write enable signal WE~ to an active state, i.e., "L" level, and transmits a status lead command "70h" to input/output terminal I/O, and sets lead enable signal RE~ to an active state, i.e., "L" level, after that (step B19).

[0158] If a status lead command "70h" is transmitted, from a flash EEPROM, reading appearance of the status data in which it is shown whether page light actuation was successful will be carried out, and it will be set to the NAND data read/write register 202.

[0159] Firmware checks whether with reference to the status data of the NAND data read/write register 202, page light actuation has been successful, and if it has succeeded, it will end light actuation (step A24). When it fails, alternative processing with an alternative block etc. is performed.

[0160] Next, with reference to drawing 16 and drawing 17, processing of firmware required for control of the read cycle of a flash EEPROM and actuation of the NAND bus interface 19 are explained.

[0161] Drawing 16 is a flow chart explaining the processing of firmware and the actuation of the NAND bus interface 19 in a read cycle, and drawing 17 is a timing chart which shows actuation of the flash EEPROM in a read cycle.

[0162] There are the 1st read cycle which reads data per page, and the 2nd read cycle which reads ** by the data of the redundancy field of each page as read cycle of a flash EEPROM 11-1 to 11-16.

[0163] Here, the 1st read cycle explains the case where the head page of a certain block is read.

[0164] With reference to one of the NAND chip ready / busy registers 215-1 to 215-16 first corresponding to the flash EEPROM chip for a lead in the firmware performed by micro PUROSSE 14, the flash EEPROM chip for [the] a lead checks whether you are the Lady State (waiting) (step A31). If it checks that he is the Lady State, firmware will carry out the I/O light of the NAND chip number register 201, and will set the chip number corresponding to the flash EEPROM chip for a lead there (step A32).

[0165] Subsequently, firmware carries out the I/O light of the NAND chip enable control register 203, and sets CE on-command which specifies activating a chip enable input (CE) there (step A33).

[0166] At this time, the chip select signals CS1-CS16 corresponding to the chip number by which the NAND bus interface 19 was set to the NAND chip number register 201 activate one (step B31). This chip select signal is supplied to chip enable input CE~ of the chip for a lead, and the chip for a lead is set as enabling state by this.

[0167] Next, firmware carries out the I/O light of the NAND command latch enabling control register 204, and sets the CLE on-command which specifies activating a command latch enabling (CLE) signal there (step A34).

[0168] The NAND bus interface 19 answers the set of a CLE on-command, and sets the command latch enable signal CLE to an active state, i.e., "H" level, (step B32).

[0169] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets a lead command "00h" there (step A35). The NAND bus interface 19 answers the set of a lead command "00h", and sets write enable signal WE~ to an active state, i.e., "L" level, and transmits a lead command "00h" to input/output terminal I/O (step B33).

[0170] Firmware carries out the I/O light of the NAND address latch enabling control register 205, and sets an ALE on-command there (step A36). The NAND bus interface 19 answers the set of an ALE on-command, and sets address latch enable signal ALE to an active state, i.e., "H" level, (step B34).

[0171] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets a memory address (column address A7-A0 and page address A20-A8) there (step A37). The NAND bus interface 19 answers the set of the address, write enable signal WE~ is set to an active state, i.e., "L" level, and the set memory address is transmitted to input/output terminal I/O (step B35).

[0172] A transfer of this memory address performs page lead actuation in the flash EEPROM for a lead. In page lead actuation, reading appearance of 256 bytes of user data stored in the page specified by the memory address, 2 bytes of LWC data, and 6 bytes of ECC data is carried out, and it is transmitted to a data register. During this page lead actuation period, the ready/busy signal from that chip (RDY/BSY) will return from a busy condition to a ready state, if a busy condition is shown and page lead actuation is completed.

[0173] If completion of page lead actuation is checked according to the content of the NAND chip ready / the busy register corresponding to the chip which carried out read access (step A38), firmware will set an ECC check initiation command to the ECC start control register 206 (step A39), will execute a MUBU string instruction after that, and will start 256 bytes of user data transfer from a data buffer 131 to a data buffer 131 (step A40).

[0174] In this data transfer, reading appearance of 256+8 bytes of data currently held at the data register of a flash EEPROM is serially carried out per 1 byte synchronizing with lead enable signal RE~ outputted from the NAND bus interface 19.

[0175] The NAND bus interface 19 performs the check operation for error detection to the data by which reading appearance is carried out from a flash EEPROM (step B36), and sets the check result to the ECC data registers 209-214 (step B37).

[0176] Firmware will investigate whether normal data have been led with reference to the content of the ECC data registers 209-214, if termination of an ECC check operation is checked (step A41). When it is detected that the error has occurred to lead data, firmware performs the error correction of 256 bytes of user data transmitted to the data buffer 131 on a data buffer 131 using the ECC check result of an operation set to the ECC data registers 209-214.

[0177] Thus, in a read cycle, an ECC check is automatically performed with the NAND bus interface 19 at the time of the data transfer from a flash EEPROM to a data buffer 131.

[0178] By the way, each of a flash EEPROM 11-1 to 11-16 has the page increment function. This page increment function is a function which starts the page lead of following *-JI automatically, when data read-out for 1 page from a data register is completed. For this reason, after serial data read-out of the head page mentioned above is completed, a flash EEPROM will be in a busy condition again, and starts the page lead of a 2 *-JI eye.

[0179] Therefore, about data read-out after the 2nd page, firmware and the NAND bus interface 19 do not need to perform the above-mentioned steps A31-A37 and the address cycle of B31-B35, and should just repeat the above-mentioned steps A38-A41, and B36 and B37, respectively.

[0180] Next, the 2nd read cycle explains the actuation in the case of reading only the data of a redundancy field.

[0181] The being the same difference of most actuation of the firmware and the NAND bus interface 19 required when performing this 2nd read cycle is the following two points with the case of the 1st read cycle.

[0182] (1) In step A35 of drawing 16, firmware publishes a command "50h" instead of a command "00h."

[0183] (2) The sequence of a read cycle is completed by 2 bytes of data read-out from a data register.

[0184] For this reason, although LWC data can be read only by performing the 2nd read cycle once, in reading ECC data, it is necessary to repeat the 2nd read cycle 3 times.

[0185] In addition, in the 2nd read cycle, it can be determined by the value of the column address set up in an address cycle from which location of a data register data are read.

[0186] Next, with reference to drawing 18 and drawing 19, processing of firmware required for control of the block erasion cycle of a flash EEPROM and actuation of the NAND bus interface 19 are explained.

[0187] Drawing 18 is a flow chart explaining the processing of firmware and the actuation of the NAND bus interface 19 in a block erasion cycle, and drawing 19 is a timing chart which shows actuation of the flash EEPROM in a block erasion cycle.

[0188] Here, it explains taking the case of the case where a certain 1-block specific content of storage is eliminated.

[0189] With reference to one of the NAND chip ready / busy registers 215-1 to 215-16 corresponding to the flash EEPROM chip with which the firmware performed by micro PUROSSE 14 includes the block for erasion first, the flash EEPROM chip for [the] erasion checks whether you are the Lady State (waiting) (step A51). If it checks that he is the Lady State, firmware will carry out the I/O light of the NAND chip number register 201, and will set the chip number corresponding to the flash EEPROM chip for erasion there (step A52).

[0190] Subsequently, firmware carries out the I/O light of the NAND chip enable control register 203, and sets CE on-command which specifies activating a chip enable input (CE) there (step A53).

[0191] At this time, the chip select signals CS1-CS16 corresponding to the chip number by which the NAND bus interface 19 was set to the NAND chip number register 201 activate one (step B51). This chip select signal is supplied to chip enable input CE~ of the chip for erasion, and the chip for erasion is set as enabling state by this.

[0192] Next, firmware carries out the I/O light of the NAND command latch enabling control register 204, and sets the CLE on-command which specifies activating a command latch enabling (CLE) signal there (step A54).

[0193] The NAND bus interface 19 answers the set of a CLE on-command, and sets the command latch enable signal CLE to an active state, i.e., "H" level, (step B-52).

[0194] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets an erasion command "60h" there (step A55). The NAND bus interface 19 answers the set of an erasion command "60h", and sets write enable signal WE~ to an active state, i.e., "L" level, and transmits an erasion command "60h" to input/output terminal I/O (step B53).

[0195] Firmware carries out the I/O light of the NAND address latch enabling control register 205, and sets an ALE on-command there (step A56). The NAND bus interface 19 answers the set of an ALE on-command, and sets address latch enable signal ALE to an active state, i.e., "H" level, (step B54).

[0196] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets a memory address (block-address A20-A12) there (step A57). The NAND bus interface 19 answers the set of the address, write enable signal WE~ is set to an active state, i.e., "L" level, and the set memory address is transmitted to input/output terminal I/O (step B55).

[0197] Then, after firmware publishes a CLE on-command, it carries out the I/O light of the NAND data read/write register 202, and sets an erasion initiation command "D0h" there (step A58). First, the NAND bus interface 19 sets write enable signal WE~ as an active state, after setting the command latch enable signal CLE as an active state, and it transmits an erasion initiation command "D0h" to input/output terminal I/O (step B56).

[0198] A transfer of this erasion initiation command "D0h" carries out automatic activation of block erasion actuation and its verification actuation in the flash EEPROM for erasion.

[0199] In block erasion actuation, only the content of storage of the block specified by the block address is eliminated. During this block erasion actuation period, the ready/busy signal from that chip (RDY/BSY) will return from a busy condition to a ready state, if a busy condition is shown and block erasion actuation is completed.

[0200] If completion of block erasion actuation is checked according to the content of the NAND chip ready / the busy register corresponding to the chip which erased (step A59), firmware will carry out the I/O light of the NAND data read/write register 202, and will set a status lead command "70h" there (step A60). The NAND bus interface 19 answers the set of a status lead command "70h", sets write enable signal WE~ to an active state, i.e., "L" level, and transmits a status lead

command "70h" to input/output terminal I/O, and sets lead enable signal RE~ to an active state, i.e., "L" level, after that (step B57).

[0201] If a status lead command "70h" is transmitted, from a flash EEPROM, reading appearance of the status data in which it is shown whether block erasion actuation was successful will be carried out, and it will be set to the NAND data read/write register 202.

[0202] Firmware checks whether with reference to the status data of the NAND data read/write register 202, block erasion actuation has been successful, and if it has succeeded, it will end an erasion cycle (step A61). When it fails, alternative processing with an alternative block etc. is performed.

[0203] In addition, if steps A54-A57 are repeated and performed and two or more block addresses are set up before issuance of a block erasion initiation command "D0h", multi-block actuation which performs two or more blocks erasion one by one can also be performed.

[0204] Next, with reference to the flow chart of drawing 20, the light control action of the RAM disk equipment 10 whole of drawing 1 performed when a light command is published from a host system is explained. This light control action is performed by using various kinds of management information tables mentioned above.

[0205] If a host system sets a light command to the command register 176 of the external bus interface 17, as for the firmware performed by the microprocessor 14, the light command will be received first (step A71). Subsequently, firmware starts the light data transfer from a host system to a data buffer 131 (step A72). This data transfer is performed through the data register 173 of the external bus interface 17.

[0206] Since access of the RAM disk equipment 10 by the host system is performed per sector, the light data for at least 1 sector are stored in a data buffer 131.

[0207] Subsequently, firmware reads in the sector number register 171, the cylinder register 174, and the drive / head register 175 the host address for disk accessing specified by the host system, i.e., a sector number, a cylinder number, and a head number, respectively, and changes it into the address for accessing a flash EEPROM 11-1 to 11-16 (step A73). In this address translation processing, as drawing 12 explained, an address translation table 132 is used. the address translation information on this address translation table 132 -- the time of starting of equipment 10 -- setting -- a flash EEPROM 11-1 to 11-16 -- respectively -- since -- it is built based on the address translation information on the led address translation table.

[0208] The chip number for light access and the memory address in the chip for light access (A20-A0) are called for as a result of address translation processing.

[0209] Next, firmware progresses to the processing (steps A74-A78) for creating the light data for 1 block to the block (a light block being called hereafter) for [which is specified in 9 bits (A20-A12) of high orders of a memory address (A20-A0)] light access.

[0210] Here, firmware secures the header table area for a light block on RAM13 first (step A74). Next, firmware leads LWC data from the head page of a light block (step A75). In lead processing of this LWC data, the 2nd read cycle using the command "50h" mentioned above is performed, and reading appearance only of the LWC data is carried out by this from the head page of a light block. This LWC data is stored in the empty area or the data buffer 131 of RAM13.

[0211] Then, firmware investigates whether "involvement elimination" data exist in a light block (step A76). Here, "involvement elimination" data mean the data as which rewriting is not demanded by the host system in the data which exist in a light block. For this reason, although "involvement elimination" data do not exist when the writing to all the pages within a light block is required, when writing is started from intermediate *-JI within a light block, or when there is less data length of light data than 1 block (eight sectors), "involvement elimination" data will exist.

[0212] When "involvement elimination" data exist, firmware leads the page which contains "involvement elimination" data from a light block by performing the 1st read cycle which used the command "00h" (step A77). The page containing this "involvement elimination" data is stored in the empty area and the data buffer 131 of RAM13.

[0213] Subsequently, firmware creates the header table corresponding to the block data for 1 block which should be written in a light block based on the RAM address which shows the storing location on the led "involvement elimination" RAM 13 in which the RAM address which shows the storing location on RAM13 of data, and the light data from a host system are stored.

[0214] That is, firmware combines data actually, and does not generate the block data for 1 block, but creates the header table 133 which aligns in order of the page number and holds the address on "involvement elimination" the address on RAM13 of data and the data buffer 131 of light data, and builds imagination block data by this (step A78).

[0215] Subsequently, firmware updates the value of the LWC data by counting up the value of the LWC data led from the light block +one time (step A79). If digit going up (carry) does not occur in the value of LWC data at this time, firmware performs light access processing to a light block (steps A80 and A81).

[0216] In this light access processing, the block erasion cycle explained by drawing 18 to the light block is performed first, and all the contents of the light block are eliminated. Subsequently, the program cycle explained by drawing 14 to all the pages of a light block is performed. In this program cycle, firmware sets the updated LWC data to the LWC register 208 of the NAND bus interface 19 first, subsequently to the header table 133, reads the data for 1 block from RAM13 per 256 bytes in order of the address currently held, and transmits it to a flash EEPROM. Thus, firmware can perform data transfer by using the header table 133, without distinguishing light data and "involvement elimination" data.

[0217] ECC data are automatically added to 256 bytes each of data transmitted with the NAND bus interface 19, and it is written in the page to which the light block of the flash EEPROM for a light corresponds.

[0218] On the other hand, when digit going up occurs in the value of UWC data, firmware performs the following processings, in order to examine whether swapping actuation is performed.

[0219] That is, firmware leads a MBI table block from the flash EEPROM for a light by performing the 1st read cycle first (step A82). The content of the led MBI table block is stored in the empty area of RAM13.

[0220] Subsequently, firmware searches the led MBI table block, detects the UWC data corresponding to a light block, updates the value of the UWC data, and gets for it blocked and counts it up +one time (step A83). In the MBI table block, as drawing 8 explained, UWC data align in order of a physical block address, and are held. For this reason, firmware can find the UWC data corresponding to a light block easily by using as an index the physical block address which shows a light block.

[0221] The value of the updated UWC data is reflected in a MBI table block and count of rewriting managed table 134 both sides.

[0222] Then, firmware carries out the sequential comparison of the value of the UWC data of the updated light block with the value of the minimum UWC data of the chip for a light, and the value of the minimum UWC data of each of other chip (step A84).

[0223] The value of the minimum UWC data currently held as a value of the minimum UWC data of the chip for a light at the swap order table within the MBI table block led from the chip for a light is used. Without this performing processing in which sorting of the count of rewriting managed table 134 on RAM13 is carried out to the small order of UWC data, although the value of the minimum UWC data of the chip for a light is got to know easily, it can do.

[0224] Moreover, as a value of the minimum UWC data of each of other chip, the value of the count of rewriting managed table 134 on RAM13 is used. By using the value of this count of rewriting managed table 134, the need of carrying out the sequential lead of the MBI table block disappears from each of all other chips other than the chip for a light, and efficient swapping examination processing is realized.

[0225] Subsequently, based on the comparison result of step A84, it investigates whether firmware is more than the threshold (alpha) the difference of the value of the UWC data and the minimum UWC data of the chip for a light with which the light block was updated was beforehand decided to be (step A85), and with [the difference] alpha [more than], swapping actuation within the chip for a light is performed (step A86).

[0226] In this swapping actuation in a chip, firmware leads the block for swapping-ed from the chip for a light first. in this case, the block for swapping-ed -- a minimum of -- it is the block with the value of UWC and the physical block address of this block is specified by the swapping-ed priority NO.1 block number on the swap order table contained in the MBI table block led at step A82. Therefore, without performing sorting of UWC data, firmware can determine the block for swapping-ed easily, and can recognize the physical block address of the block for [the] swapping-ed.

[0227] Next, firmware is between a light block and the block for swapping-ed, and replaces the host address. In this case, the address translation information on MBI table block ***** led from the address translation table 132 and the chip for a light on RAM132 is updated.

[0228] Next, firmware performs light access for writing the content of the swapping-ed block sucked up on RAM13 for exchange of the content of storage between the light block in the flash EEPROM for a light, and the block for swapping-ed in the light block of the flash EEPROM for a light, and light access for writing the light data for 1 block on RAM13 in the swapping-ed block of a flash EEPROM. Then, firmware performs light access for writing in the content of the MBI table block with which it was updated on RAM13 by the MBI table block in the flash EEPROM for a light for renewal of the MBI table block in the flash EEPROM for a light. These three light accesses are performed with the same gestalt as the case where step A81 explains, respectively.

[0229] Next, processing when the difference of the value of the UWC data and the minimum UWC data of the chip for a light with which the light block was updated in step A86 is under alpha is explained.

[0230] In this case, it investigates whether firmware is more than the threshold (alpha+beta) the difference of the value of the UWC data with which the light block was updated, and the minimum UWC data of one of other chips was beforehand decided to be (step A87), and with [that difference] alpha+beta [more than], swapping actuation between the chip for a light and a chip with the value of that minimum UWC data is performed (step A88).

[0231] In the swapping actuation covering between this chip, firmware leads a MBI table block on RAM13 from a chip with the block for swapping-ed first, and recognizes the swapping-ed priority NO.1 block number on the swap order table contained in that MBI table block.

[0232] Subsequently, firmware leads the block for swapping-ed from the chip for swapping-ed using the physical block address which swapping-ed priority NO.1 block number shows.

[0233] Next, firmware is between a light block and the block for swapping-ed, and replaces the host address. In this case, each address translation information on the address translation table 132 on RAM132, the MBI table block led from the chip for a light, and the MBI table block led from the chip for swapping-ed is updated.

[0234] Next, firmware performs light access for writing the content of the light block in the flash EEPROM for a light, and the swapping-ed block sucked up on RAM13 for exchange of the content of storage during the block for swapping-ed of the chip for swapping-ed in the light block of the flash EEPROM for a light, and light access for writing the light data for 1 block on RAM13 in the swapping-ed block of the flash EEPROM for swapping-ed. Then, firmware performs light access for writing in the content of the MBI table block with which it was updated on RAM13 by the MBI table block in the flash EEPROM for a light, and light access for writing in the content of the MBI table block with which it was updated on RAM13

by the MBI table block in the flash EEPROM for swapping-ed, in order to update the MBI table block of each flash EEPROM for swapping-ed [for a light / flash EEPROM]. These four light accesses are performed with the same gestalt as the case where step A81 explains, respectively.

[0235] Thus, SUWABBINGU actuation is performed between chips within a chip, and, as for the case of the swapping between chips, read access actuation and light access actuation are mostly needed from the case of the swapping in a chip by a unit of 1 time, respectively.

[0236] However, in this example, since the threshold ($\alpha + \beta$) for determining whether perform swapping or not is set as the bigger value than the threshold (α) in the case of performing swapping actuation in a chip, it can control the occurrence frequency of the swapping between chips compared with the swapping in a chip. Therefore, the degradation by frequent occurrence of the swapping actuation between chips can be prevented.

[0237] When both the differences of the count of replacement are smaller than a predetermined threshold at step A85 and **** A87, firmware performs light access for writing the MBI table block containing the UWC data updated at step A83 in the flash EEPROM for a light (step A89), and, subsequently performs light access processing to a light block. These light access is performed with the same gestalt as the case where step A81 explains, respectively.

[0238] As mentioned above, since light access is immediately performed, without making examination of swapping processing when the digit riser of LWC data does not occur, the time amount of the whole which data writing takes rather than the case where a digit riser occurs can be shortened.

[0239] The value of LWC data is counted up every [1 / +] from the initial value at every light access, and when it amounts to 1K times, a digit riser generates it. For this reason, it is desirable to use not the same value but a discrete value different, respectively as initial value of the LWC data written in all blocks included in a flash EEPROM 11-1 to 11-16 at the time of disk manufacture and shipment. In this case, since the stage when a digit riser occurs shifts for every block even if light access to a contiguous block occurs continuously, the situation of the examination processing and swapping processing of swapping resulting from digit going up concentrating, and generating can be prevented.

[0240] As explained above, in this example, ECC data are generated by the ECC arithmetic circuit 21 for every page, and it is written in the same page as the data stream from which that ECC data was set as the object of an ECC operation. For this reason, even if, even if a two or more pages defect cel occurs in each same bit position, the defect cel contained in the data stream of a flash EEPROM set as the object of an ECC operation is set only to one. Therefore, without using the complicated high ECC operation expression of data bounce-back capability, by the easy usual ECC operation, it becomes possible to perform error detection and correction, and the RAM disk equipment 10 in which rapid access is possible is realized highly [storage dependability].

[0241] Moreover, LWC data are stored in the redundancy field of the head page of each block as count information of rewriting corresponding to the block, and the value of the LWC data is updated in the case of the data writing to the block for a light. The updated LWC data is added to the light data which should be written in a head page, and is transmitted to the data register 112 of a flash EEPROM. Thus, since LWC data are managed within each block, only by light access to the same block, renewal of data writing and LWC data can be performed, and improvement in the light access engine performance can be aimed at as a result.

[0242]

[Effect of the Invention] As explained above, according to this invention, it becomes possible by managing ECC and the count information of rewriting using the redundancy field of each page of a flash EEPROM to realize high life-ization of the dependability of enough stored data, and a flash EEPROM.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

TECHNICAL FIELD

[Industrial Application] Especially this invention builds in the flash EEPROM which has two or more page fields where each includes a data storage area and a redundancy field about RAM disk equipment, and relates to the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] Many of information processors, such as the conventional workstation and a personal computer, used the magnetic disk drive as secondary storage. While a magnetic disk drive has an advantage, like a bit unit price with the high dependability of record is cheap, it has a fault, such as being weak, in a physical impact with the great size of equipment.

[0003] Then, attentions have gathered for RAM disk equipment strong also against an impact with it in recent years. [the small size of equipment, and] [physical] The flash EEPROM whose RAM disk equipment is the semiconductor memory of the non-volatile in which package elimination is possible electrically is used as secondary storage, such as a personal computer, like the conventional magnetic disk drive etc. Since a part for mechanical moving part like the magnetic head of a magnetic disk drive or a revolution disk is not included in this RAM disk equipment, it is hard to generate malfunction and failure by the physical impact. Moreover, there is an advantage, like the size as equipment also becomes small.

[0004] By the way, in RAM disk equipment, the ECC calculation function is prepared, and ECC corresponding to the content of data is usually generated and added to the data written in a flash EEPROM. By using this ECC calculation function, the dependability of the data written in a flash EEPROM can be raised.

[0005] In the former which shows the typical data storage format adopted with conventional RAM disk equipment with an ECC calculation function function to drawing 21 , ECC is generated for every sector which is a disk-accessing unit by host equipment. For this reason, ECC is stored in the location which follows that sector data, when sector data with the size of 512 bytes are continued and stored in 2 pages of a flash EEPROM as shown in drawing 21 . Although determined by the physical size of a flash EEPROM, when in what page of a flash EEPROM sector data are continued and stored uses the 16M bit flash EEPROM with the largest size by which current development is carried out, 512 bytes of sector data will be continued and stored in 2 pages of a flash EEPROM 1 like a graphic display.

[0006] In this case, the sector data covering those 2 pages calculate, and ECC is generated. This generated ECC follows sector data and is stored in the 2nd page.

[0007] However, in the conventional RAM disk equipment which used such a data storage format, consideration of the following error generating properties which a flash EEPROM has produces the nonconformity of the dependability of data storage falling.

[0008] That is, in the semiconductor memory which makes a flash EEPROM the start, the error mode in which a defect occurs simultaneously exists in two or more memory cells connected not only to a certain specific memory cell but to the same bit line. This error mode is caused by the defect of a bit line, and the poor contact between a bit line and a cel.

[0009] When such error mode occurs, a defect cel is generated in the same bit position of each two or more pages of a flash EEPROM as shown in drawing 21 . In this case, in conventional RAM disk equipment, a two-place error will be generated like a graphic display at the data stream set as the object of an ECC operation.

[0010] Although detection and correction of the error can generally be easily performed by the error detection and correction using ECC when an error is one place, about two or more errors, the detection and correction become difficult. In order to correspond to such an error, the complicated high ECC operation expression of data bounce-back capability is needed.

[0011] However, if ECC operation expression is complicated, while the configuration of RAM disk equipment will be complicated, in order for generation of ECC at the time of the data light and the ECC check operation at the time of a data lead to take much time amount, the fault that the access rate of RAM disk equipment falls is caused.

[0012] Moreover, with conventional RAM disk equipment, all not only of generation of ECC and an ECC check operation but the error detection based on the result of an operation and error correction processing were performed by hardware. For this reason, there was a problem that the gate number for realizing that hardware increased, or a read/write rate fell as a result by complication of hardware.

[0013] Furthermore, with conventional RAM disk equipment, package management of the count of rewriting of each block of a flash EEPROM is carried out by the count information of rewriting stored in the specific block of the flash EEPROM. For this reason, in case data are written in a flash EEPROM, two light accesses with light access for updating light access for data writing and the count information of rewriting are needed. Therefore, while the time amount which data writing takes became long, the count of light access to the block which has memorized the count information of rewriting projected and increased compared with other blocks, and there was a problem to which the life of a flash EEPROM becomes short by this.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention] As explained above, according to this invention, it becomes possible by managing ECC and the count information of rewriting using the redundancy field of each page of a flash EEPROM to realize high life-ization of the dependability of enough stored data, and a flash EEPROM.

[Translation done.]

* NOTICES *

JP0 and NCIP1 are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] With conventional RAM disk equipment, the physical structure of a flash EEPROM was not fully taken into consideration. For this reason, even if it prepared the ECC calculation function, sufficient storage dependability was not able to be acquired. Moreover, since the count information of rewriting was concentrated and stored in the specific block, only the count of rewriting of the block increased and there was a fault to which the life of a flash EEPROM becomes short by this.

[0015] This invention was made in view of such a point, uses the redundancy field of each page of a flash EEPROM effectively, and enables it to manage ECC and the count information of rewriting, and it aims at offering the RAM disk equipment which can realize high life-ization of the dependability of enough stored data, and a flash EEPROM.

[Translation done.]

*** NOTICES ***

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

OPERATION

[Means for Solving the Problem and its Function] The memory cell array in which invention concerning claim 1 has two or more pages in which each includes a data storage area and a redundancy field, Have a data register holding the data for 1 page, and the flash EEPROM by which data transfer between a data register and a memory cell array is performed per page is built in. In the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment The data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, A light demand is answered from said host equipment, and the light access means which carries out light access of said flash EEPROM is provided. This light access means The light data stored in said data buffer are calculated in the data stream unit equivalent to the size of the data storage area of each of said page. So that it may be written in the data storage area and redundancy field an error correcting code generation means to generate an error correcting code for every data stream, and whose error correcting code corresponding to said data stream and it are the same pages It is characterized by including a means to add the error correcting code corresponding to it for every data stream, and to transmit to the data register of said flash EEPROM.

[0017] In this RAM disk equipment, an error correcting code is generated for every page of a flash EEPROM, and it is written in the redundancy field that error correcting code of whose is each page. For this reason, even if, even if a two or more pages defect cel occurs in each same bit position, the defect cel contained in the data stream of a flash EEPROM set as the operation object of an error correcting code is set only to one. Therefore, without using the complicated high ECC operation expression of data bounce-back capability, by the easy usual ECC operation, it becomes possible to perform error detection and correction, and the RAM disk equipment in which rapid access is possible is realized highly [storage dependability].

[0018] Moreover, in invention concerning claim 3 and claim 4, it is characterized by writing not only an error correcting code but the bad sector information specified from host equipment and error information in a redundancy field, respectively.

[0019] By this, since it becomes unnecessary to carry out the centralized control of bad sector information or the error information, it becomes unnecessary to prepare the exclusive field for storing these information, and the data storage area of a flash EEPROM can be used effectively for storing of user data.

[0020] Moreover, invention concerning claim 5 is characterized by being constituted so that ROM in which the program performed by said host equipment was stored may be provided in addition to the configuration of claim 1, the lead demand of the predetermined address from said host equipment may be answered and a program may be transmitted to said host equipment from said ROM.

[0021] In this RAM disk equipment, ROM in which the program of an application program, an operating system, etc. other than a flash EEPROM was stored is prepared, and when the lead demand of a predetermined address is published from host equipment, reading appearance of the program of this ROM is carried out. Since rewriting does not generate programs, such as an application program, the data storage area of an expensive flash EEPROM can be used effectively for storing of user data by storing the program in comparatively cheap ROM.

[0022] Moreover, the memory cell array in which invention concerning claim 6 has two or more pages in which each includes a data storage area and a redundancy field, Have a data register holding the data for 1 page, and the flash EEPROM by which data transfer between a data register and a memory cell array is performed per page is built in. In the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment A microprocessor and the data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, The flash memory interface circuitry which is controlled by said microprocessor and carries out read/write access of said flash EEPROM, The I/O register group in which read/write is possible is provided by said microprocessor. Said flash memory interface circuitry The light data by which reading appearance is carried out in the data stream unit which is equivalent to the size of the data storage area of each of said page from said data buffer with said microprocessor are calculated. So that it may be written in the error correcting code generation circuit which generates an error correcting code for every data stream, and the data storage area and redundancy field whose error correcting code corresponding to said data stream and it is the same page The light data transfer circuit which adds the error correcting code corresponding to it for every data stream, and is transmitted to the data register of said flash EEPROM, The lead data transfer circuit which takes out the data stream stored in said data storage area from the lead data by which reading appearance is carried out from the data register of said flash EEPROM, and is transmitted to said data

buffer, Error detection of said lead data is performed using the error correcting code contained in said lead data. Said microprocessor is characterized by performing the error correction of the lead data transmitted to said data buffer according to the result of an operation set to said I/O register including the error correcting code checking circuit which sets the detection result to said I/O register.

[0023] The error correction of lead data is performed in this RAM disk equipment by the firmware which the check for generation of an error correcting code and error detection is performed by the hardware logic of a flash memory interface circuitry, and is performed by the microprocessor. Therefore, a load distribution is efficiently made by firmware and hardware and the configuration of hardware can be simplified rather than the case where all of the ECC check operation for generation of ECC and error detection and an error correction are performed by hardware.

[0024] Moreover, invention concerning claim 7 includes two or more blocks which have two or more pages in which each includes a data storage area and a redundancy field, respectively. The memory cell array by which the count information of rewriting which shows the count of rewriting of the block is stored in the redundancy field of the predetermined page of each block, Have a data register holding the data for 1 page, and data transfer between a data register and a memory cell array is performed per page. Elimination actuation builds in the flash EEPROM performed per block. It is RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment. The data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, A light demand is answered from said host equipment, and the light access means which carries out light access of said flash EEPROM is provided. This light access means A means to read the count information of rewriting from said predetermined page within the block for a light specified by said light demand, and to update the value of the count information of rewriting, Add said updated count information of rewriting to the light data which should be written in said predetermined page, and page data are generated. It is characterized by providing a means to transmit said page data to the data register of said flash EEPROM so that it may be stored in the data storage area and redundancy field said light data and said whose updated count information of rewriting are said predetermined pages.

[0025] In this RAM disk equipment, the count information of rewriting corresponding to that block is stored in the redundancy field of the predetermined page of each block, and the value of the count information of rewriting on that block for a light is updated in the case of the data writing to the block for a light. The updated count information of rewriting is added to the light data which should be written in a predetermined page, and is transmitted to the data register of a flash EEPROM. Since the count information of rewriting is managed within each block, it becomes unnecessary thus, to carry out the centralized control of the count information of rewriting in a specific field. For this reason, generating of the problem that the count of rewriting of a specific region projects and increases by light access for updating the count information of rewriting can be prevented. Moreover, only by light access to the same block, renewal of data writing and the count information of rewriting can carry out, and improvement in the light access engine performance can be aimed at as a result.

[0026] Moreover, invention concerning claim 8 includes two or more blocks which have two or more pages in which each includes a data storage area and a redundancy field, respectively. The memory cell array by which the count information of rewriting which shows the count of rewriting of the block is stored in the redundancy field of the predetermined page of each block, Have a data register holding the data for 1 page, and a command is answered from the outside. The elimination actuation which eliminates the content of storage per block, the light actuation which writes data in a memory cell array per page, And the flash EEPROM which performs lead actuation which reads data from a memory cell array in a page unit is built in. In the RAM disk equipment which accesses said flash EEPROM according to the disk-accessing demand from host equipment The data buffer which stores the lead data by which reading appearance was carried out from the light data transmitted from said host equipment, and said flash EEPROM, A light demand is answered from said host equipment, and the light access means which carries out light access of said flash EEPROM is provided. This light access means Publish a lead command and said flash EEPROM is set as a lead mode of operation. A means to read the count information of rewriting from said predetermined page within the block for a light specified by said light demand, and to update the value of the count information of rewriting, When the page of the outside for light access exists in the block for a light, The data stored in the page of the outside for [the] light access are read on said data buffer. The storing location on the data buffer of each data which constitutes the block data for 1 block which should be written in the block for [said] a light based on the address on the data buffer of the read data, and the address of the light data stored in said data buffer A means to generate the header table holding the shown address information, and a means to publish an erasion command, to set said flash EEPROM as an elimination mode of operation, and to eliminate the content of storage of said block for a light, With reference to the address information of said header table, the light data which should be written in said predetermined page are read from said data buffer. So that said updated count information of rewriting may be added to the light data, page data may be generated and it may be stored in the data storage area and redundancy field said light data and said whose updated count information of rewriting are said predetermined pages A means to transmit said page data to the data register of said flash EEPROM, With reference to the address information of said header table, the light data which should be written in in addition to said predetermined page are read from said data buffer one by one, and it is characterized by including a means to transmit it to the data register of said flash EEPROM.

[0027] In this RAM disk equipment, when the page of the outside for light access exists within the block for a light, reading appearance of the so-called involvement elimination data stored in the page of the outside for light access is carried out, and a header table is generated based on the address of each of that involvement elimination data and light data from a host. If a header table is generated, after the content of storage of the block for a light is eliminated, according to the address

information of a header table, block data will be transmitted to a data register in a page unit from a data buffer, and write-in actuation will be performed within a flash EEPROM. In this data transfer, the count information of rewriting is added and transmitted about the data which should be written in a predetermined page. For this reason, like the RAM disk equipment of claim 7, only by light access to the same block, renewal of data writing and the count information of rewriting can carry out, and improvement in the light access engine performance can be aimed at as a result. Moreover, data transfer processing to a flash EEPROM from a data buffer can be performed, without distinguishing involvement elimination data and light data by referring to this header table, since the header table holding the address information which shows the storing location of the block data for 1 block is generated.

[0028] Moreover, in addition to the configuration of claim 8, it is the configuration of having had error correcting code generation and the calculation function of claim 1, and, as for invention of claim 9, an error correcting code is stored in the redundancy field to which the count information of rewriting and an error correcting code are stored, and are each of other page in the redundancy field of the predetermined page of each block. In this RAM disk equipment, it becomes possible to use a redundancy field for validity more.

[0029] Moreover, invention of claim 10 carries out another **** management of the count information of rewriting on each block in the configuration of claim 7 at the high-order-bit section and lower bit section. It is the configuration of concentrating and storing the 1st count data of rewriting which consists of the high-order-bit sections in a predetermined block, and distributing and storing in the redundancy field of a predetermined page the 2nd count data of rewriting which consists of the lower bit sections for every block.

[0030] In this RAM disk equipment, since only the low order bit data of the count information of rewriting are stored in a redundancy field, the activity field of a redundancy field can be reduced. Moreover, whenever rewriting occurs, it is necessary to update low order bit data but, and since this updating is performed by light access for the data writing to the block for rewriting, and coincidence, it does not need to perform separately light access for data writing, and light access for renewal of the count information of rewriting like the RAM disk equipment of claim 7. Moreover, the count of rewriting of all blocks can be managed only by data read-out from one block, without reading the count information of rewriting sequentially from all blocks, since the data of the high-order-bit section of the count information of rewriting are concentrated and stored in the predetermined block.

[0031] Moreover, invention of claim 11 establishes the swapping means for equalizing the count of rewriting to the count management tool of rewriting of claim 10. In this RAM disk equipment, when a digit riser occurs to the data of the lower bit section of the count information of rewriting, it is examined whether the block exchange for leveling is performed. When the block with which the difference of the count of rewriting with the block for a light reaches beyond a predetermined value exists, the block is determined as a swapping point block, and the address and the content of storage are mutually replaced between the block and the block for a light. Thus, since it is examined whether the block exchange for leveling is performed on condition that the digit riser of the lower bit section data of the count information of rewriting, it becomes unnecessary to monitor the count of rewriting of each block continuously, and management of the count of rewriting becomes easy.

[0032] Moreover, invention of claim 12 stores a respectively discrete value for every block as initial value of the 2nd count data of rewriting in the time of disk manufacture and shipment etc. It can prevent the examination processing and swapping processing of swapping resulting from digit going up concentrating, and generating by this, even if rewriting to two or more blocks occurs continuously, since the stage when a digit riser occurs shifts for every block.

[0033] Moreover, in addition to the 1st count data of rewriting, the swapping table is stored in the predetermined block of FUSSHU EEPROM in invention of claim 13. The priority information which shows the priority of a swapping point block in this swapping table according to the value of the 1st count data of rewriting is held. Therefore, it becomes possible to specify a swapping point block easily, without carrying out sorting of the 1st count data of rewriting.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

EXAMPLE

[Example] Hereafter, the example of this invention is explained with reference to a drawing.

[0035] The configuration of the RAM disk equipment concerning one example of this invention is shown in drawing 1. This RAM disk equipment 10 is used as an alternative of a hard disk drive unit, connecting with host systems, such as a personal computer, emulates the disk-accessing demand from a host system (driver for a file system and magnetic disk drives) to the access request to a flash EEPROM, and accesses the flash EEPROM of RAM disk equipment 10 built-in. To this RAM disk equipment 10, it is possible to carry a maximum of 16 flash EEPROM chips. These flash EEPROMs are mounted in the circuit base of RAM disk equipment 10 free [attachment and detachment] through an IC socket, respectively. In drawing 1, the condition that 16 NAND mold flash EEPROMs 11-1 to 11-16 are carried is shown. A controller 12 and RAM13 are further formed in RAM disk equipment 10.

[0036] A flash EEPROM 11-1 to 11-16 is used as a record medium of this RAM disk equipment 10, and is equivalent to the magnetic-recording medium of a hard disk drive unit. These flash EEPROMs 11-1 to 11-16 are connected to the controller 12 through the NAND memory bus 14. The various control signal lines connected to the data bus of 8-bit width of face and the control signal terminal of each flash EEPROM are defined as the NAND memory bus 14. Common connection of an I/O data bus and the various control signal lines is made at data input/output terminal I/O of a flash EEPROM 11-1 to 11-16, and the control signal input terminal CONT (the command latch enabling terminal CLE, the address latch enabling terminal ALE, the write enable terminal WE, the lead enabling terminal RE, etc. are included), respectively.

[0037] Furthermore, between the flash EEPROM 11-1 to 11-6, and the controller 12, the chip select signal (CS1-CS8) line and the Ready/Busy signal line (R/B1 - R/B16) are independently arranged for every chip, respectively.

[0038] These flash EEPROMs 11-1 to 11-6 are the 16M bit NAND molds EEPROM, respectively, and they are equipped with the memory cell array 111 and the data register 112 as shown in drawing 2. The memory cell array 111 has a 8K line x264 train x8 bit bit pattern, and has 512 blocks. Data elimination can be performed in this block unit. Each block consists of 16 pages (line), and each page is equipped with 256 bytes of data storage area, and 8 bytes of redundancy field. Writing and read-out of data are performed per page through 256+8 bytes of data register 112. The redundancy field of each page is used for storing of ECC or the count data of rewriting in this RAM disk equipment.

[0039] The flash EEPROM 11-1 to 11-6 has modes of operation, such as a write mode, a Read mode, and washout mode, respectively, and these modes of operation are specified by the command from a controller 12.

[0040] In a Read mode, automatic activation of the data transfer (page lead) of the page unit from the memory cell array 111 to a data register 112 is carried out inside a flash EEPROM. a Ready/Busy signal line is set as a Busy condition during this data transfer period -- having -- this -- a page lead -- a working thing is notified to a controller 12. Reading appearance of the data for 1 page transmitted to the data register 112 is serially carried out by 8 bitwises from input/output terminal I/O0 - I/O7. Termination of the serial lead of the data for 1 page performs the page lead of the following page automatically.

[0041] In a write mode, automatic activation of the data transfer (page light) of the page unit from the data register 112 to the memory cell array 111 is carried out inside a flash EEPROM. a Ready/Busy signal line is set as a Busy condition during this data transfer period -- having -- this -- a page light -- a working thing is notified to a controller 12.

[0042] In washout mode, package elimination of the content of storage of a block of the arbitration specified by the command from a controller is carried out.

[0043] In RAM disk equipment 10, most processings for control of these flash EEPROMs 11-1 to 11-6 are controlled by firmware. The ECC check in the case of the ECC generation in the case of the data writing to a flash EEPROM and data read-out from a flash EEPROM is performed by hardware.

[0044] Next, the configuration of the disk controller 12 of RAM disk equipment 10 is explained.

[0045] The disk controller 12 is realized by one LSI, and accumulation formation of a microprocessor (MPU) 14, the processor bus interface 15, ROM16, the external bus interface 17, the buffer memory controller 18, and the NAND bus interface 19 is carried out at the LSI chip as shown in drawing 1.

[0046] A microprocessor 14 is what performs firmware stored in ROM16 and controls actuation of this RAM disk equipment 10 whole. The host address for disk accessing sent from CPU1 of a personal computer The address translation processing changed into the memory address for accessing a flash EEPROM 11-1 to 11-16, the access-control processing for carrying out read/write access of the flash EEPROM 11-1 to 11-16, and a flash EEPROM 11-1 to 11-6 -- control of the swapping processing for equalizing management of each count of data rewriting and the count of data rewriting etc. is performed.

[0047] Various programs, such as firmware performed by the microprocessor 14 and an operating system performed by CPU1 of a personal computer, and an application program, are stored in ROM16. the operating system and the application program are managed as a file within ROM16, and these programs are published from CPU1 to RAM disk equipment 10 -- reading appearance is carried out, and reading appearance is carried out from ROM16 according to a demand, and it is loaded to the main memory 3 of a personal computer.

[0048] The continuous disk address is mapped by ROM16 and the flash EEPROM 11-1 to 11-16, and ROM16 is also used for them as some storages of a disk unit 10. For this reason, read-out of the operating system and application program which are stored in ROM16 is performed only when the read-out demand from CPU1 specifies a certain specific address, and when specifying other addresses, read access to a flash EEPROM 11-1 to 11-16 is performed.

[0049] The external bus interface 17 is an interface based on IDE or a PCMCIA specification, and is connected to the system bus 2 of a personal computer. The I/O register group for the communication link with CPU1 is prepared in the external bus interface 17. The sector number register 171, the sector count register 172, a data register 173, the cylinder register 174, the drive / head register 175, the command register 176, the status register 177, etc. are contained in the register group. Read/write is possible for these registers respectively by the microprocessor 14 and CPU1 of a personal computer.

[0050] The light of the sector number which specifies an access head location is carried out to the sector number register 171 by CPU1. The light of the number of sectors for read/write is carried out to the sector count register 172 by CPU1. The lead data by which reading appearance is carried out from the light data or the RAM disk equipment 10 supplied from CPU1 are set to a data register 173. The light of the cylinder number for read/write is carried out to the cylinder register 174 by CPU1. The light of the drive number for read/write and the head number is carried out to a drive / head register 175 by CPU1. The light of a lead command, a light command, etc. which specify actuation of RAM disk equipment 10 is carried out to a command register 176 by CPU1. The various statuses of the RAM disk equipment 10 which should be notified to CPU1 are set to a status register 177.

[0051] The buffer memory controller 18 is for carrying out the access control of RAM13, and performs data transfer between a system bus 2 and RAM13 etc. There is data transfer between others, RAM13, flash EEPROM 11-1 - 11 -16 [data transfer / between a system bus 2 and RAM13] among the data transfer. The latter data transfer is performed by activation of the MUBU string instruction by the microprocessor 14.

[0052] The NAND bus interface 19 is hardware logic which carries out the access control of the flash EEPROM 11-1 to 11-16 to the bottom of control of a microprocessor 14, and delivers and receives the data transfer between flash EEPROMs 11-1 to 11-16, and the various control signals between flash EEPROMs. The I/O register group 20 in which read/write is possible, and the ECC arithmetic circuit 21 are established in this NAND bus interface 19 by the microprocessor 14. The ECC arithmetic circuit 21 performs the operation for the operation for ECC generation, and an ECC check.

[0053] In the I/O register group 20, as shown in drawing 3 The NAND chip number register 201, the NAND data read/write register 202, the NAND chip enable control register 203, the NAND command latch enabling control register 204, the NAND address latch enabling control register 205, The ECC start control register 206, ECC data JISUTA 209-214 of an ECC ready / busy status register 207, and 208 or 6 LWC registers, the 16 NAND chip readies / busy status registers 215-1 to 215-16, etc. are contained.

[0054] The NAND chip number register 201 is a register for choosing to which NAND mold flash EEPROM processing is performed, and the value which shows the chip number of the NAND mold flash EEPROM for access is set by the processor 14.

[0055] The NAND data read/write register 202 is a register for performing read/write of the data to a flash EEPROM through a data bus, and is used for issuance of the command to a flash EEPROM, issuance of the address, the lead of status data, etc.

[0056] The NAND chip enable control register 203 is a register for controlling the chip enable terminal CE of a flash EEPROM, and the flag which shows generating / un-generating is set by the processor 14. [of a chip select signal] If a flag "1" is set to a bit 0, the NAND bus interface 19 will generate the chip select signal (CS) for activating the chip enable terminal CE of the flash EEPROM chosen with the NAND chip number register 201.

[0057] The NAND command latch enabling control register 204 is a register for controlling the command latch enabling terminal CLE of a flash EEPROM, and the flag which specifies whether the command latch rice ruble terminal CLE is activated is set by the processor 14. If a flag "1" is set to a bit 0, the command latch enabling terminal CLE of a flash EEPROM will be actively set up with the NAND bus interface 19. Thereby, it is notified to a data bus to a flash EEPROM that the command was published.

[0058] The NAND address latch enabling control register 205 is a register for controlling the address latch enabling terminal ALE of a flash EEPROM, and the flag which specifies whether the address latch rice ruble terminal ALE is made into the bitter taste eve is set by the processor 14. If a flag "1" is set to a bit 0, the address latch enabling terminal ALE of a flash EEPROM will be actively set up with the NAND bus interface 19. Thereby, it is notified to a data bus to a flash EEPROM that the address was published.

[0059] The ECC start control register 206 is a register for controlling activation initiation of an ECC operation, and before starting the data transfer between RAM13 and a flash EEPROM 11-1 to 11-16, the flag which specifies activation initiation of an ECC operation is set by the processor 14. "1" is set to a bit 0 when performing data transfer from a flash EEPROM to RAM13. In this case, the ECC arithmetic circuit 21 performs an ECC check about the data by which reading appearance is carried out from a flash EEPROM. "0" is set to a bit 0 when performing data transfer from RAM13 to a flash EEPROM. In this case, the ECC arithmetic circuit 21 generates ECC about the data transmitted from RAM13.

[0060] An ECC ready / busy status register 207 is status registers which the ECC operation (ECC generation or ECC check) by the ECC arithmetic circuit 21 shows whether it is under [activation] ***** . The NAND bus interface 19 sets the flag of "0" to a bit 0, after setting the flag of "1" to a bit 0 and completing activation of an ECC operation, if it is among ECC instruction execution. If a processor 14 investigates the existence of termination of ECC data processing with reference to the flag of a register 207 after 256 bytes of data transfer termination between RAM13 and a flash EEPROM 11-1 to 11-16 and termination is checked, it will start 256 bytes of next data transfer processing.

[0061] The data which should be stored in 2 bytes of the head of 8 bytes of redundancy field of a flash EEPROM, for example, the LWC (Low Write Count) data which are the lower bit of the count data of rewriting, are set to the LWC register 208. In this case, the LWC data corresponding to the block for a light of a flash EEPROM which serves as the destination from RAM13 in advance of that data transfer in the data transfer to a flash EEPROM are set by firmware. In the data transfer from a flash EEPROM to RAM13, LWC data are taken out from the data stream by which reading appearance is carried out from a flash EEPROM with the NAND bus interface 19, and it is set.

[0062] The ECC data registers 209-214 are the register groups for holding 6 bytes of ECC data. At the time of the data transfer from RAM13 to a flash EEPROM, 6 bytes of ECC data corresponding to 256 bytes of light data are generated by the ECC arithmetic circuit 21, and the generated ECC data is set to the ECC data registers 209-214. At the time of the data transfer from a flash EEPROM to RAM13, the result of the ECC check operation performed by the ECC arithmetic circuit 21 is set to registers 209-214. It performs, in order that this ECC check operation may perform error detection. In this ECC check operation, 6 bytes of ECC data corresponding to 256 bytes of lead data are generated again, and the comparison with 6 bytes of ECC data by which reading appearance was actually carried out to 6 bytes of that ECC data from the flash EEPROM is performed per cutting tool. It will be set to "0" and the bit 0 of the register with which "1" corresponds if inharmonious if in agreement. Therefore, if the error has not occurred, all the values set to the bit 0 of the ECC data registers 209-214 are set to "0."

[0063] a NAND chip ready / busy status register 215-1 to 215-16 -- 16 flash EEPROMs 11-1 to 11-16 -- it is a register group holding each ready/busy status, and the flag which shows whether a corresponding flash EEPROM is a ready state or it is a busy condition is set to the bit 0 of each register with the NAND interface 19. If it is a ready state, as for that, a flag "0" will be set for a flag "1" by the busy condition. It is judged by the ready/busy signal (R/B1 - R/B16) independently inputted from a flash EEPROM 11-1 to 11-16, respectively for every chip whether it is a ready state or it is a busy condition.

[0064] RAM13 of drawing 1 consists of dynamic RAMs, and a part of the storage region is used as a data buffer 131 holding the light data transmitted from CPU1 of a personal computer, and the lead data by which reading appearance is carried out from a flash EEPROM 11-1 to 11-16. Moreover, RAM13 is used as a working area of a microprocessor 14, and the count of rewriting managed table 134 for managing the address translation table 132 for changing the host address into a memory address, the header table 133 which manages the storing location of the block data for 1 block which should be written in a flash EEPROM, and the count of rewriting of a flash EEPROM 11-1 to 11-16 per block etc. is stored here.

[0065] Next, with reference to drawing 4, the memory map in the RAM disk equipment 10 managed by the microprocessor 14 is explained.

[0066] The RAM field, the NAND memory data area, the application ROM (APL-ROM) field, and the firmware ROM (FW-ROM) field are mapped like a graphic display by the memory address space which a microprocessor 14 manages.

[0067] A RAM field is the memory address space for accessing RAM13, and the data buffer 131 etc. is assigned here.

[0068] A NAND memory data area is 256 bytes of address space for accessing a flash EEPROM 11-1 to 11-16. 256 bytes of data transfer is performed by the MUBU string instruction of a microprocessor 14 in read/write access of the data to a flash EEPROM 11-1 to 11-16. For this reason, 256 bytes of address space is secured as a NAND memory data area for 256 bytes of this data transfer. By using this NAND memory data area, all data transfer between a data buffer 131 and a flash EEPROM 11-1 to 11-16 can be performed as processing of firmware.

[0069] next, drawing 5 -- referring to -- a flash EEPROM 11-1 to 11-16 -- the utilization gestalt of each block is explained.

[0070] The block of a flash EEPROM 11-1 to 11-16 is used not only for storing of user data but for storing of the various management information for controlling these flash EEPROMs. The assignment of the user data and management information to a flash EEPROM 11-1 to 11-16 is as follows.

[0071] That is, in the NAND mold flash EEPROM 11-1 of chip number #0, 504 blocks of block 0 - block 503 are used as memory block for storing user data in 512 blocks like a graphic display. Moreover, block 504 is used as a MBI (MemoryBlock Infomation) table block which memorizes the management information for managing the address, a count of rewriting, etc. of 512 blocks in a chip. Moreover, blocks 505-509 are used as a preliminary block for the alternative, when a defect occurs in memory block and a MBI table block. Block 511 is used as a KONFIGU block which memorizes the configuration information of RAM disk equipment 10, and block 510 is used as a preliminary block for an alternative of a KONFIGU block.

[0072] Only one KONFIGU block exists by the equipment 10 whole, and the information which shows the number of chips of the flash EEPROM carried in RAM disk equipment 10, the MBI table block in each flash EEPROM existence location, etc. is managed here.

[0073] In the NAND mold flash EEPROM 11-2 to 11-16 of chip number #1-#16, like a graphic display, 504 blocks of block 0 - block 503 are used as memory block for storing user data, and block 504 is used as a MBI (Memory Block Infomation) table block which memorizes the management information for managing the address and the count of rewriting of 512 blocks in a chip. Moreover, blocks 505-511 are used as a block for an alternative of memory block and a MBI table block.

[0074] Although the physical block structure of memory block, a MBI table block, an alternative block, and each KONHIIGU block is as drawing 2 having explained, the data storage format within these blocks is explained hereafter.

[0075] The data storage format of memory block for storing user data is shown in drawing 6.

[0076] As drawing 2 explained, the NAND mold flash EEPROM used with this RAM disk equipment 10 is a 16M bit chip, and each block consists of 16 pages from a page 0 to a page 15. Each page consists of 256 bytes of a data area, and 8 bytes of a redundancy field.

[0077] Generally, disk accessing by the host system is performed per 512 bytes of sector. In this case, the size of 256 bytes of user data stored in the data area of each page serves as half [of the user data size for 1 sector]. In this RAM disk equipment 10, only 256 bytes of data area is used for storing of user data, and the redundancy field which is 8 bytes is not used for storing of user data. Therefore, 512 bytes of user data are continued and stored in 2 pages followed within a block in this RAM disk equipment 10. For this reason, the user data stored in 1 block turn into the data for 8 sector, for example, the data from a sector 0 to a sector 7.

[0078] That is, to 256 bytes in the first half of a sector 0, and a page 1, 256 bytes in the second half of a sector 1 are stored in a page 0 in 256 bytes in the first half of a sector 1, and a page 3 at 256 bytes in the second half of a sector 0, and a page 2, and user data are stored per 1 / 2 sectors like the following as shown in drawing 5.

[0079] pages 0-15 -- 8 bytes of each redundancy field is divided into 2 bytes of field, and 6 bytes of field, and ECC is stored in 6 bytes of the field. This ECC corresponds to 256 bytes of user data stored in the data area of the same page.

[0080] Namely, ECC stored in the redundancy field of a page 1 is similarly generated from 256 bytes of user data stored in the data area of a page 1 by generating ECC stored in the redundancy field of a page 0 from 256 bytes of user data stored in the data area of a page 0.

[0081] Thus, in this RAM disk equipment 10, ECC is generated not per 512 bytes of sector unit but per 256 bytes of page, and the data stream and ECC for an operation are stored in the same page. For this reason, even if the error of a flash EEPROM that a two or more pages defect cel occurs in each same bit position occurs like a graphic display, the defect cel contained in the data stream set as the operation object of ECC will be set only to one. Therefore, it becomes possible to perform error detection and correction by the easy usual ECC operation, without using the complicated high ECC operation expression of data bounce-back capability.

[0082] Moreover, LWC data are stored in the remaining 2 bytes of the redundancy field of a page 0 as block status information of block 0. This LWC data consists of data of the 10 bits of lower bit sections of the count data of rewriting in which the count of rewriting of block 0 is shown, for example, the low order of the count data of rewriting of 26-bit width of face.

[0083] one LWC data is stored for every block -- having -- block 1 - block 503 -- LWC data are similarly stored in 2 bytes of the head of the redundancy field of each head page as the block status information.

[0084] Furthermore, BAD sector information and ECC error information are stored in the remaining 2 bytes of the redundancy field of a page 1 as sector status information of a sector 0.

[0085] BAD sector information is status information the sector 0 indicates it to be whether it is a bad sector. the same -- the three-page page 5 and a page 15 -- the remaining 2 bytes of each redundancy field -- the 1 sector sector 2 and .. a sector 7 - - it is alike, respectively and the BAD sector information that it corresponds is stored. These BAD(s) sector information is given by CPU1 of the personal computer, when the personal computer which is the host system of RAM disk equipment 10 executes the formatting command for formatting RAM disk equipment 10.

[0086] ECC error information is status information which shows the existence of ECC error generating to the sector. For this ECC error information, a personal computer is WRITE. When the data writing by the LONG command is performed and the error of the burst error used as an ECC error etc. occurs, it is given by CPU1 of a personal computer.

[0087] Next, with reference to drawing 7, the data storage format of a MBI table block is explained.

[0088] The MBI table block consists of one block containing 16 pages from a page 0 to a page 15. Each page consists of 256 bytes of a data area, and 8 bytes of a redundancy field. ECC corresponding to 256 bytes of data stored in the data area of the page is stored in 6 bytes of the back of 8 bytes of the redundancy fields. Moreover, 2 bytes of the head of a redundancy field are intact.

[0089] The address translation table, the UWC table, and the swap order table are stored in the data area of a MBI table block like a graphic display.

[0090] The address translation table in a MBI table block holds the address translation information for changing the physical block addresses 0-511 of the blocks 0-511 in a chip into the host address to which it is specified with a cylinder number, a head number, and a sector number. The host address is stored in order of the physical block address.

[0091] Here, as a value of the host address in a MBI table block, the host logical address calculated from the cylinder number given from a host system, a head number, and not the sector number itself but these cylinder numbers, the head number, and the sector number is used.

[0092] This host logical address is the serial number of the sector at the time of assigning the sector number which continues sequentially from "0" to all the sectors (= 8x512x16 sectors) prepared in a flash EEPROM 11-1 to 11-16. Therefore, the bit width of face of the host logical address becomes 16 bits. However, there is no address translation information managed by the address translation table of this MBI table block per 1 sector, and since it is for performing address translation per 8 sectors which are the number of sectors contained in 1 block, only 13 bits of high orders in the 16-bit host logical address are actually stored on the table.

[0093] A UWC table holds the UWC (UPPER WRITECOUNT) data which consist of the high-order-bit sections of the count data of rewriting in which the count of rewriting of each 512 blocks in a chip is shown. UWC data have 16-bit width of face, and if this UWC data and LWC of the above-mentioned 10-bit width of face are doubled, they can count it to rewriting of 64 million counts at the maximum. The value of UWC data is counted up +one time, whenever a digit riser occurs to corresponding LWC data (i.e., whenever 1024 rewritings occur).

[0094] These address translation tables and a UWC table are stored in the data area of a page 5 to the page 12 of a MBI table block like a graphic display. An address translation table and a UWC table are transmitted to RAM13 at the time of equipment starting, i.e., the power-on of RAM disk equipment 10, and reside permanently as an address translation table 132 and count of rewriting managed tables 134, respectively.

[0095] A swap order table is a table referred to by the microprocessor 14 in the case of the swapping actuation for block exchange, and holds the block address which shows the block (swapped block) of the swapping point in order of a priority (order with the small value of UWC data). The swap order table is stored in the data area of a page 13 to the page 15 of a MBI table block like a graphic display. A swap order table does not reside in RAM13 permanently, but at the time of examination of block exchange, i.e., digit riser generating to UWC data from LWC data, it is transmitted to RAM13 and refer to it for a microprocessor 14.

[0096] The configuration of an address translation table and a UWC table is shown in drawing 8.

[0097] Like a graphic display, the host address and the storing field of UWC data have the data size of 2 bytes, respectively. Here, since the host address is 13 bits as mentioned above, the high order triplet (a bit 15 - bit 13) of 2 bytes of storing field is not used. The host address and UWC data are arranged by turns in order of the physical block address.

[0098] That is, the host address and UWC data are stored in order by 64 blocks (block 0 - block 63) from the head in 256 bytes of data area of the page 5 in a MBI table block in the sequence of the UWC data corresponding to the UWC data corresponding to the host address corresponding to the physical block address 0, and the physical block address 0, the host address corresponding to the physical block address 1, and the physical block address 1.

[0099] Similarly, the host address and UWC data are stored in order by 64 blocks (block 448 - block 511) from the head in 256 bytes of data area of a page 12 in the sequence of the UWC data corresponding to the UWC data corresponding to the host address corresponding to the physical block address 448, and the physical block address 448, the host address corresponding to the physical block address 449, and the physical block address 449.

[0100] The configuration of a swap order table is shown in drawing 9.

[0101] The swap order table is divided into the 1st order table corresponding to 256 blocks of the first half from block 0 to block 255, and the 2nd order table corresponding to 256 blocks of the second half from block 256 to block 511.

[0102] It sets on the 1st order table and the value of the minimum UWC in block 255 is stored in 2 bytes of field (addresses 800h and 801h) of the head from the block 0.

[0103] the following 1 byte (address 802h) -- a minimum of -- the value which shows the total of the block with the value of UWC is stored.

[0104] Furthermore, the value which shows the total of the block with the value of minimum UWC+1 is stored in the following 1 byte (address 803h). here -- "minimum UWC+1" -- to the last -- a minimum of -- the value which added 1 to UWC -- being shown -- **** -- a minimum of -- small UWC is not shown in the degree of UWC. A block total with the value of minimum UWC+1 may serve as zero.

[0105] Henceforth, sequential storing of the data in which a block total with the value of minimum UWC+120 is shown is carried out from the data in which a block total with the value of minimum UWC+2 is shown at address 804h-87Ah.

[0106] The data in which the count of rewriting of the MBI table block [itself] is shown are stored in address 87Ch-87Fh.

[0107] The information as an original swap order table is stored in address 880h-97Fh. namely, -- as the value which shows the block for [of whenever / of the highest priority / swapped] to address 880h in block 0 - block 255 -- a minimum of -- the physical block address of a block which has the value of UWC is stored. When two or more blocks which have the value of the minimum UWC exist, priority becomes [a physical block address] high from the rearmost block at order. Hereafter, the block address of the block for a swap is similarly stored in 97Fh(s) in order of a priority from address 881h.

[0108] Next, the 2nd order table is explained.

[0109] the 2nd order table -- it is and the value of the minimum UWC in block 511 is stored in 2 bytes of field (addresses 980h and 981h) of the head from the block 256 except for the MBI table block, the alternative block, and the KONHIIGU block. That is, since, as for an alternative block etc., the count of rewriting is not advancing at all, the count of rewriting of these blocks is disregarded. Moreover, when an alternative occurs actually, suppose the value of the UWC data of the alternative block that UWC of a defective block is inherited not the actual count of rewriting of the alternative block but a substituting agency that is,. This is for preventing the degradation by frequent occurrence of swapping actuation. If the value of the actual UWC data of an alternative block is adopted as it is, the value of the count of rewriting projects, a small block will appear suddenly and the swapping actuation which sets the alternative block as the object-ed [swapping] will be generated repeatedly.

[0110] the following 1 byte (address 982h) -- a minimum of -- the value which shows the total of the block with the value of UWC is stored.

[0111] Furthermore, the value which shows the total of the block with the value of minimum UWC+1 is stored in the following 1 byte (address 983h). Henceforth, sequential storing of the data in which a block total with the value of minimum UWC+120 is shown is carried out from the data in which a block total with the value of minimum UWC+2 is shown at

address 984h-9FAh.

[0112] The information as an original swap order table is stored in address A00h-AFFh. namely, -- as the value which shows the block for [of whenever / of the highest priority / swapped] to address A00h in block 256 - block 511 -- a minimum of -- the physical block address of a block which has the value of UWC is stored. When two or more blocks which have the value of the minimum UWC exist, priority becomes [a physical block address] high from the rearmost block at order. Hereafter, the block address of the block for a swap is similarly stored in AFFh in order of a priority from address A01h.

[0113] Thus, in the swap order table, the block used as the swapping point is stored in order of the priority. For this reason, it becomes possible to specify the swapping point, without a microprocessor 14 performing sorting of UWC data by referring to this swap order table.

[0114] The data storage format of a KONHIIGU block is shown in drawing 10.

[0115] The KONHIIGU block consists of one block containing 16 pages from a page 0 to a page 15. Each page consists of 256 bytes of a data area, and 8 bytes of a redundancy field. ECC corresponding to 256 bytes of data stored in the data area of the page is stored in 6 bytes of 8 bytes of redundancy field. Moreover, the remaining 2 bytes of a redundancy field are intact.

[0116] KONHIIGU information, CIS information, and drive ID information are stored in the data area of a KONHIIGU block like a graphic display.

[0117] KONHIIGU information defines the physical block address which shows the number of chips of the flash EEPROM carried in this RAM disk equipment 10, the number of the maximum chips which can be written in simultaneous, and the MBI table block in each chip existence location. This KONHIIGU information is referred to by the microprocessor 14 at the time of the power-on of RAM disk equipment 10, and the initialization process according to that KONHIIGU information is performed.

[0118] CIS information is card attribute information notified to a host system, when using RAM disk equipment 10 as a card of a PCMCIA specification. The information about physical specifications, such as a class of memory chip, capacity, and access time, and the information about the application which shows that this card is a disk unit etc. are defined as this card attribute information. This CIS information is transmitted to RAM13 by the microprocessor 14 at the time of the power-on of RAM disk equipment 10. The CIS information transmitted to RAM13 answers the read-out demand of the predetermined address from a host system, and is transmitted to a host system.

[0119] Drive ID information is information which shows the drive configuration of RAM disk equipment 10, and the total number of sectors logically assigned to a built-in flash EEPROM, the number of cylinders, the number of heads, the number of sectors per one track, etc. are defined here. This drive ID information answers a drive discernment command from a host system, and is transmitted to a host system by the microprocessor 14.

[0120] Thus, in this example, ECC corresponding to the data stored in the data area of that page is stored in the redundancy field of each page also in the MBI table block and KONHIIGU block which are used not only for memory block for storing user data but for management information storing. Therefore, also in these MBI(s) table block and a KONHIIGU block, it becomes possible to perform error detection and correction by the easy usual ECC operation, without using the complicated high ECC operation expression of data bounce-back capability.

[0121] Next, with reference to drawing 11, the configuration of the address translation table 132 prepared in RAM13 is explained.

[0122] a thing for this address translation table 132 to change the host address specified with a cylinder number, a head number, and a sector number into the physical memory address for accessing a flash EEPROM 11-1 to 11-116 -- it is -- this address translation table 132 -- a flash EEPROM 11-1 to 11-116 -- it is generated based on the address translation information on each MBI table block.

[0123] In this address translation table 132, a 13-bit block sector address is used as the host address. This block sector address consists of 13 bits of high orders in the host logical address (16 bits) which shows the serial number of the sector calculated from the cylinder number, the head number, and the sector number, as mentioned above. Moreover, a physical memory address consists of the 4-bit chip number address and a physics block address in [of 9 bits] a chip. The 4-bit chip number address specifies one of 16 chips from the flash EEPROM 11-1 of chip number #0 to the flash EEPROM 11-6 of chip number #15. The physics block address in [of 9 bits] a chip specifies one in 512 blocks of the block 0 in the chip specified by the chip number address - block 511. The physical memory address (the chip number address, physics block address in a chip) is stored in order of the host address.

[0124] Next, with reference to drawing 12, the principle of the address translation actuation using an address translation table 132 is explained.

[0125] First, the cylinder number, head number, and sector number which were specified from the host system are changed into the host logical address of 16-bit width of face by the firmware performed by the microprocessor 14. Subsequently, an address translation table 132 is searched by making into an index the block sector address which consists of 13 bits of high orders of the host logical address of the 16-bit width of face, and the chip number address of the 4-bit width of face corresponding to a block sector address and the physical block address of 9-bit width of face are determined.

[0126] The chip number address is set to the NAND chip number register 201 of the NAND bus interface 19 by firmware for a chip select. Moreover, the sector address within a block which consists of the low order triplet of the host logical address of 16-bit width of face, the even-/recto appointed address of 1-bit width of face, and the column address of 8-bit width of face are added to a 9-bit physical block address within NAND BAIINTAFESU 19, and the memory address (A20-A0) of 20-bit width of face is generated. This memory address (A20-A0) is set to the NAND data read/write register 202 of the NAND bus

interface 19 for addressing. Since access of the RAM disk equipment 10 performed by the host system is performed per sector in order to carry out read/write of the user data, both of even-/recto appointed addresses (A8) of 1-bit width of face and column addresses (A7-A0) of 8-bit width of face are usually oar "0."

[0127] Next, with reference to drawing 13, ECC generation / check actuation of the ECC arithmetic circuit 21 of the NAND bus interface 19 is explained.

[0128] First, generation actuation of ECC is explained with reference to drawing 13 (a).

[0129] Reading appearance of the user data stored in the data buffer 131 of RAM13 at the time of the data writing to the NAND mold flash EEPROM 11-1 to 11-16 is carried out by the microprocessor 14 per 256 bytes, and they are transmitted to the data register of a flash EEPROM through the NAND bus interface 19.

[0130] In the NAND bus interface 19, 256 bytes of user data are sent to the ECC arithmetic circuit 21, and the operation for generating ECC there is performed. Here, Hamming code etc. is used as ECC.

[0131] An ECC operation is performed 256 bytes of not only user data but for 258 bytes of data which consist of the 256 bytes of user data and data which should be stored in the first 2 bytes of a redundancy field actually.

[0132] For example, in the data writing to the head page of each block, 258 bytes of data which consist of 256 bytes of user data and 2 bytes of LWC data are set as the object of an ECC operation. 2 bytes of this LWC data is set to the LWC register 208 by the microprocessor 14 in advance of 256 bytes of user data transfer.

[0133] The ECC arithmetic circuit 21 calculates 258 bytes of data which consist of 256 bytes of user data, and 2 bytes of LWC data set to the LWC register 208 per predetermined data stream, and carries out sequential generation of 6 bytes of Hamming code ECC. These Hamming codes ECC are held temporarily at the ECC data registers 209-214.

[0134] If generation processing of ECC is completed, the ECC arithmetic circuit 21 will be made to follow 256 bytes of user data, and will transmit 2 bytes of LWC data, and 6 bytes of ECC data to the data register of a flash EEPROM. Thereby, the data for 1 page which consists of 256 bytes of user data, 2 bytes of LWC data, and 6 bytes of ECC data are set to the data register of a flash EEPROM.

[0135] Next, check actuation of ECC is explained with reference to drawing 13 (b).

[0136] The data for 1 page by which reading appearance is carried out from a flash EEPROM at the time of data read-out from the NAND mold flash EEPROM 11-1 to 11-16 are sent to the NAND bus interface 19.

[0137] In the NAND bus interface 19, 256 bytes of user data are picked out from the data for 1 page, and it is sent to the data buffer 131 of RAM13. Moreover, the data for 1 page is sent to the ECC arithmetic circuit 21.

[0138] First, the ECC arithmetic circuit 21 picks out LWC data from the data for 1 page, sets it to the LWC register 208, and performs the ECC check operation for error detection after that.

[0139] In this ECC check operation, the ECC arithmetic circuit 21 calculates 258 bytes of data which consist of 256 bytes of user data, and 2 bytes of LWC data in the same data stream unit as the case of ECC generation processing, and carries out sequential generation of 6 bytes of Hamming code ECC. And Hamming code ECC contained in the data for 1 page by which reading appearance was carried out from the generated Hamming code ECC and flash EEPROM is compared per cutting tool, and the comparison result which shows coincidence/inequality is set to the ECC data registers 209-215. When there is no error generating, the data "0" in which coincidence is shown are altogether set to the ECC data registers 209-215. If the content of the ECC data registers 209-215 was referred to by the microprocessor 14 and the error has occurred, the error correction of the data on a data buffer 131 will be performed by the microprocessor 14.

[0140] Next, in order to make clear the functional assignment of the firmware performed by micro PUSESSA 14 and the NAND bus interface 19, the access-control actuation to a flash EEPROM is explained.

[0141] First, with reference to drawing 14 and drawing 15, processing of firmware required for control of the program cycle of a flash EEPROM and actuation of the NAND bus interface 19 are explained.

[0142] Drawing 14 is a flow chart explaining the processing of firmware and the actuation of the NAND bus interface 19 in a program cycle, and drawing 15 is a timing chart which shows actuation of the flash EEPROM in a program cycle.

[0143] Here, 256 bytes of user data, 2 bytes of LWC data, and 6 bytes of ECC data are explained to the head page of a certain block taking the case of the case where a light is carried out.

[0144] With reference to one of the NAND chip ready / busy registers 215-1 to 215-16 first corresponding to the flash EEPROM chip for a light in the firmware performed by micro PUROSSE 14, the flash EEPROM chip for [the] a light checks whether you are the Lady State (waiting) (step A11). If it checks that he is the Lady State, firmware will carry out the I/O light of the NAND chip number register 201, and will set the chip number corresponding to the flash EEPROM chip for a light there (step A12).

[0145] Subsequently, firmware carries out the I/O light of the NAND chip enable control register 203, and sets CE on-command which specifies activating a chip enable input (CE) there (step A13).

[0146] At this time, the chip select signals CS1-CS16 corresponding to the chip number by which the NAND bus interface 19 was set to the NAND chip number register 201 activate one (step B11). This chip select signal is supplied to chip enable input CE~ of the chip for a light, and the chip for a light is set as enabling state by this.

[0147] Next, firmware carries out the I/O light of the NAND command latch enabling control register 204, and sets the CLE on-command which specifies activating a command latch enabling (CLE) signal there (step A14).

[0148] The NAND bus interface 19 answers the set of a CLE on-command, and sets the command latch enable signal CLE to an active state, i.e., "H" level, (step B12).

[0149] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets a serial data input

command "80h" there (step A15). The NAND bus interface 19 answers the set of a serial data input command "80h", and sets write enable signal WE~ to an active state, i.e., "L" level, and transmits a serial data input command "80h" to input/output terminal I/O (step B13).

[0150] Firmware carries out the I/O light of the NAND address latch enabling control register 205, and sets an ALE on-command there (step A16). The NAND bus interface 19 answers the set of an ALE on-command, and sets address latch enable signal ALE to an active state, i.e., "H" level, (step B14).

[0151] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets a memory address (column address A7-A0 and page address A20-A8) there (step A17). The NAND bus interface 19 answers the set of the address, write enable signal WE~ is set to an active state, i.e., "L" level, and the set memory address is transmitted to input/output terminal I/O (step B15).

[0152] Firmware sets to the LWC register 208 the LWC data which read from the block for a light beforehand and were counted up +one time before this program cycle activation (step A18). And firmware sets an ECC initiation command to the ECC start control register 206 (step A19), and transmits 256 bytes of user data to a flash EEPROM from a data buffer 131 by activation of a MUBU string instruction after that (step A20).

[0153] The NAND bus interface 19 answers an ECC initiation command, starts the operation for ECC generation, and generates 6 bytes of ECC data corresponding to 256 bytes of user data, and LWC data (step B16). And the NAND bus interface 19 follows 256 bytes of user data, and starts 2 bytes of LWC data, and 6 bytes of ECC data transfer (step B17). Thereby, 256 bytes of user data, 2 bytes of LWC data, and 6 bytes of ECC data are transmitted as serial input data (DIN) for 1 page as shown in the data register of the flash EEPROM for a light at drawing 15.

[0154] Then, after firmware publishes a CLE on-command, it carries out the I/O light of the NAND data read/write register 202, and sets a program command "10h" there (step A21). First, the NAND bus interface 19 sets write enable signal WE~ as an active state, after setting the command latch enable signal CLE as an active state, and it transmits a program command "10h" to input/output terminal I/O (step B18).

[0155] A transfer of this program command "10h" carries out automatic activation of page light actuation and the verification actuation in the flash EEPROM for a light.

[0156] In page light actuation, page data transfer for 1 page from a data register to a memory cell array is performed, and 256 bytes of user data of a data register are written in the data area of the page, and 2 bytes of LWC data of a data register and 6 bytes of ECC data are written in the redundancy field of the page. During this page write-in actuation period, the ready/busy signal from that chip (RDY/BSY) will return from a busy condition to a ready state, if a busy condition is shown and write-in actuation is completed.

[0157] If completion of write-in actuation is checked according to the content of the NAND chip ready / the busy register corresponding to the chip which carried out light access (step A22), firmware will carry out the I/O light of the NAND data read/write register 202, and will set a status lead command "70h" there (step A23). The NAND bus interface 19 answers the set of a status lead command "70h", sets write enable signal WE~ to an active state, i.e., "L" level, and transmits a status lead command "70h" to input/output terminal I/O, and sets lead enable signal RE~ to an active state, i.e., "L" level, after that (step B19).

[0158] If a status lead command "70h" is transmitted, from a flash EEPROM, reading appearance of the status data in which it is shown whether page light actuation was successful will be carried out, and it will be set to the NAND data read/write register 202.

[0159] Firmware checks whether with reference to the status data of the NAND data read/write register 202, page light actuation has been successful, and if it has succeeded, it will end light actuation (step A24). When it fails, alternative processing with an alternative block etc. is performed.

[0160] Next, with reference to drawing 16 and drawing 17, processing of firmware required for control of the read cycle of a flash EEPROM and actuation of the NAND bus interface 19 are explained.

[0161] Drawing 16 is a flow chart explaining the processing of firmware and the actuation of the NAND bus interface 19 in a read cycle, and drawing 17 is a timing chart which shows actuation of the flash EEPROM in a read cycle.

[0162] There are the 1st read cycle which reads data per page, and the 2nd read cycle which reads ** by the data of the redundancy field of each page as read cycle of a flash EEPROM 11-1 to 11-16.

[0163] Here, the 1st read cycle explains the case where the head page of a certain block is read.

[0164] With reference to one of the NAND chip ready / busy registers 215-1 to 215-16 first corresponding to the flash EEPROM chip for a lead in the firmware performed by micro PUROSSE 14, the flash EEPROM chip for [the] a lead checks whether you are the Lady State (waiting) (step A31). If it checks that he is the Lady State, firmware will carry out the I/O light of the NAND chip number register 201, and will set the chip number corresponding to the flash EEPROM chip for a lead there (step A32).

[0165] Subsequently, firmware carries out the I/O light of the NAND chip enable control register 203, and sets CE on-command which specifies activating a chip enable input (CE) there (step A33).

[0166] At this time, the chip select signals CS1-CS16 corresponding to the chip number by which the NAND bus interface 19 was set to the NAND chip number register 201 activate one (step B31). This chip select signal is supplied to chip enable input CE~ of the chip for a lead, and the chip for a lead is set as enabling state by this.

[0167] Next, firmware carries out the I/O light of the NAND command latch enabling control register 204, and sets the CLE on-command which specifies activating a command latch enabling (CLE) signal there (step A34).

[0168] The NAND bus interface 19 answers the set of a CLE on-command, and sets the command latch enable signal CLE to an active state, i.e., "H" level, (step B32).

[0169] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets a lead command "00h" there (step A35). The NAND bus interface 19 answers the set of a lead command "00h", and sets write enable signal WE~ to an active state, i.e., "L" level, and transmits a lead command "00h" to input/output terminal I/O (step B33).

[0170] Firmware carries out the I/O light of the NAND address latch enabling control register 205, and sets an ALE on-command there (step A36). The NAND bus interface 19 answers the set of an ALE on-command, and sets address latch enable signal ALE to an active state, i.e., "H" level, (step B34).

[0171] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets a memory address (column address A7-A0 and page address A20-A8) there (step A37). The NAND bus interface 19 answers the set of the address, write enable signal WE~ is set to an active state, i.e., "L" level, and the set memory address is transmitted to input/output terminal I/O (step B35).

[0172] A transfer of this memory address performs page lead actuation in the flash EEPROM for a lead. In page lead actuation, reading appearance of 256 bytes of user data stored in the page specified by the memory address, 2 bytes of LWC data, and 6 bytes of ECC data is carried out, and it is transmitted to a data register. During this page lead actuation period, the ready/busy signal from that chip (RDY/BSY) will return from a busy condition to a ready state, if a busy condition is shown and page lead actuation is completed.

[0173] If completion of page lead actuation is checked according to the content of the NAND chip ready / the busy register corresponding to the chip which carried out read access (step A38), firmware will set an ECC check initiation command to the ECC start control register 206 (step A39), will execute a MUBU string instruction after that, and will start 256 bytes of user data transfer from a data buffer 131 to a data buffer 131 (step A40).

[0174] In this data transfer, reading appearance of 256+8 bytes of data currently held at the data register of a flash EEPROM is serially carried out per 1 byte synchronizing with lead enable signal RE~ outputted from the NAND bus interface 19.

[0175] The NAND bus interface 19 performs the check operation for error detection to the data by which reading appearance is carried out from a flash EEPROM (step B36), and sets the check result to the ECC data registers 209-214 (step B37).

[0176] Firmware will investigate whether normal data have been led with reference to the content of the ECC data registers 209-214, if termination of an ECC check operation is checked (step A41). When it is detected that the error has occurred to lead data, firmware performs the error correction of 256 bytes of user data transmitted to the data buffer 131 on a data buffer 131 using the ECC check result of an operation set to the ECC data registers 209-214.

[0177] Thus, in a read cycle, an ECC check is automatically performed with the NAND bus interface 19 at the time of the data transfer from a flash EEPROM to a data buffer 131.

[0178] By the way, each of a flash EEPROM 11-1 to 11-16 has the page increment function. This page increment function is a function which starts the page lead of following *-JI automatically, when data read-out for 1 page from a data register is completed. For this reason, after serial data read-out of the head page mentioned above is completed, a flash EEPROM will be in a busy condition again, and starts the page lead of a 2 *-JI eye.

[0179] Therefore, about data read-out after the 2nd page, firmware and the NAND bus interface 19 do not need to perform the above-mentioned steps A31-A37 and the address cycle of B31-B35, and should just repeat the above-mentioned steps A38-A41, and B36 and B37, respectively.

[0180] Next, the 2nd read cycle explains the actuation in the case of reading only the data of a redundancy field.

[0181] The being the same difference of most actuation of the firmware and the NAND bus interface 19 required when performing this 2nd read cycle is the following two points with the case of the 1st read cycle.

[0182] (1) In step A35 of drawing 16, firmware publishes a command "50h" instead of a command "00h."

[0183] (2) The sequence of a read cycle is completed by 2 bytes of data read-out from a data register.

[0184] For this reason, although LWC data can be read only by performing the 2nd read cycle once, in reading ECC data, it is necessary to repeat the 2nd read cycle 3 times.

[0185] In addition, in the 2nd read cycle, it can be determined by the value of the column address set up in an address cycle from which location of a data register data are read.

[0186] Next, with reference to drawing 18 and drawing 19, processing of firmware required for control of the block erasion cycle of a flash EEPROM and actuation of the NAND bus interface 19 are explained.

[0187] Drawing 18 is a flow chart explaining the processing of firmware and the actuation of the NAND bus interface 19 in a block erasion cycle, and drawing 19 is a timing chart which shows actuation of the flash EEPROM in a block erasion cycle.

[0188] Here, it explains taking the case of the case where a certain 1-block specific content of storage is eliminated.

[0189] With reference to one of the NAND chip ready / busy registers 215-1 to 215-16 corresponding to the flash EEPROM chip with which the firmware performed by micro PUROSSE 14 includes the block for erasion first, the flash EEPROM chip for [the] erasion checks whether you are the Lady State (waiting) (step A51). If it checks that he is the Lady State, firmware will carry out the I/O light of the NAND chip number register 201, and will set the chip number corresponding to the flash EEPROM chip for erasion there (step A52).

[0190] Subsequently, firmware carries out the I/O light of the NAND chip enable control register 203, and sets CE on-command which specifies activating a chip enable input (CE) there (step A53).

[0191] At this time, the chip select signals CS1-CS16 corresponding to the chip number by which the NAND bus interface 19 was set to the NAND chip number register 201 activate one (step B51). This chip select signal is supplied to chip enable

input CE~ of the chip for erasion, and the chip for erasion is set as enabling state by this.

[0192] Next, firmware carries out the I/O light of the NAND command latch enabling control register 204, and sets the CLE on-command which specifies activating a command latch enabling (CLE) signal there (step A54).

[0193] The NAND bus interface 19 answers the set of a CLE on-command, and sets the command latch enable signal CLE to an active state, i.e., "H" level, (step B-52).

[0194] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets an erasion command "60h" there (step A55). The NAND bus interface 19 answers the set of an erasion command "60h", and sets write enable signal WE~ to an active state, i.e., "L" level, and transmits an erasion command "60h" to input/output terminal I/O (step B53).

[0195] Firmware carries out the I/O light of the NAND address latch enabling control register 205, and sets an ALE on-command there (step A56). The NAND bus interface 19 answers the set of an ALE on-command, and sets address latch enable signal ALE to an active state, i.e., "H" level, (step B54).

[0196] Subsequently, firmware carries out the I/O light of the NAND data read/write register 202, and sets a memory address (block-address A20-A12) there (step A57). The NAND bus interface 19 answers the set of the address, write enable signal WE~ is set to an active state, i.e., "L" level, and the set memory address is transmitted to input/output terminal I/O (step B55).

[0197] Then, after firmware publishes a CLE on-command, it carries out the I/O light of the NAND data read/write register 202, and sets an erasion initiation command "D0h" there (step A58). First, the NAND bus interface 19 sets write enable signal WE~ as an active state, after setting the command latch enable signal CLE as an active state, and it transmits an erasion initiation command "D0h" to input/output terminal I/O (step B56).

[0198] A transfer of this erasion initiation command "D0h" carries out automatic activation of block erasion actuation and its verification actuation in the flash EEPROM for erasion.

[0199] In block erasion actuation, only the content of storage of the block specified by the block address is eliminated. During this block erasion actuation period, the ready/busy signal from that chip (RDY/BSY) will return from a busy condition to a ready state, if a busy condition is shown and block erasion actuation is completed.

[0200] If completion of block erasion actuation is checked according to the content of the NAND chip ready / the busy register corresponding to the chip which erased (step A59), firmware will carry out the I/O light of the NAND data read/write register 202, and will set a status lead command "70h" there (step A60). The NAND bus interface 19 answers the set of a status lead command "70h", sets write enable signal WE~ to an active state, i.e., "L" level, and transmits a status lead command "70h" to input/output terminal I/O, and sets lead enable signal RE~ to an active state, i.e., "L" level, after that (step B57).

[0201] If a status lead command "70h" is transmitted, from a flash EEPROM, reading appearance of the status data in which it is shown whether block erasion actuation was successful will be carried out, and it will be set to the NAND data read/write register 202.

[0202] Firmware checks whether with reference to the status data of the NAND data read/write register 202, block erasion actuation has been successful, and if it has succeeded, it will end an erasion cycle (step A61). When it fails, alternative processing with an alternative block etc. is performed.

[0203] In addition, if steps A54-A57 are repeated and performed and two or more block addresses are set up before issuance of a block erasion initiation command "D0h", multi-block actuation which performs two or more blocks erasion one by one can also be performed.

[0204] Next, with reference to the flow chart of drawing 20, the light control action of the RAM disk equipment 10 whole of drawing 1 performed when a light command is published from a host system is explained. This light control action is performed by using various kinds of management information tables mentioned above.

[0205] If a host system sets a light command to the command register 176 of the external bus interface 17, as for the firmware performed by the microprocessor 14, the light command will be received first (step A71). Subsequently, firmware starts the light data transfer from a host system to a data buffer 131 (step A72). This data transfer is performed through the data register 173 of the external bus interface 17.

[0206] Since access of the RAM disk equipment 10 by the host system is performed per sector, the light data for at least 1 sector are stored in a data buffer 131.

[0207] Subsequently, firmware reads in the sector number register 171, the cylinder register 174, and the drive / head register 175 the host address for disk accessing specified by the host system, i.e., a sector number, a cylinder number, and a head number, respectively, and changes it into the address for accessing a flash EEPROM 11-1 to 11-16 (step A73). In this address translation processing, as drawing 12 explained, an address translation table 132 is used. the address translation information on this address translation table 132 -- the time of starting of equipment 10 -- setting -- a flash EEPROM 11-1 to 11-16 -- respectively -- since -- it is built based on the address translation information on the led address translation table.

[0208] The chip number for light access and the memory address in the chip for light access (A20-A0) are called for as a result of address translation processing.

[0209] Next, firmware progresses to the processing (steps A74-A78) for creating the light data for 1 block to the block (a light block being called hereafter) for [which is specified in 9 bits (A20-A12) of high orders of a memory address (A20-A0)] light access.

[0210] Here, firmware secures the header table area for a light block on RAM13 first (step A74). Next, firmware leads LWC

data from the head page of a light block (step A75). In lead processing of this LWC data, the 2nd read cycle using the command "50h" mentioned above is performed, and reading appearance only of the LWC data is carried out by this from the head page of a light block. This LWC data is stored in the empty area or the data buffer 131 of RAM13.

[0211] Then, firmware investigates whether "involvement elimination" data exist in a light block (step A76). Here, "involvement elimination" data mean the data as which rewriting is not demanded by the host system in the data which exist in a light block. For this reason, although "involvement elimination" data do not exist when the writing to all the pages within a light block is required, when writing is started from intermediate **-JI within a light block, or when there is less data length of light data than 1 block (eight sectors), "involvement elimination" data will exist.

[0212] When "involvement elimination" data exist, firmware leads the page which contains "involvement elimination" data from a light block by performing the 1st read cycle which used the command "00h" (step A77). The page containing this "involvement elimination" data is stored in the empty area and the data buffer 131 of RAM13.

[0213] Subsequently, firmware creates the header table corresponding to the block data for 1 block which should be written in a light block based on the RAM address which shows the storing location on the led "involvement elimination" RAM 13 in which the RAM address which shows the storing location on RAM13 of data, and the light data from a host system are stored.

[0214] That is, firmware combines data actually, and does not generate the block data for 1 block, but creates the header table 133 which aligns in order of the page number and holds the address on "involvement elimination" the address on RAM13 of data and the data buffer 131 of light data, and builds imagination block data by this (step A78).

[0215] Subsequently, firmware updates the value of the LWC data by counting up the value of the LWC data led from the light block +one time (step A79). If digit going up (carry) does not occur in the value of LWC data at this time, firmware performs light access processing to a light block (steps A80 and A81).

[0216] In this light access processing, the block erasion cycle explained by drawing 18 to the light block is performed first, and all the contents of the light block are eliminated. Subsequently, the program cycle explained by drawing 14 to all the pages of a light block is performed. In this program cycle, firmware sets the updated LWC data to the LWC register 208 of the NAND bus interface 19 first, subsequently to the header table 133, reads the data for 1 block from RAM13 per 256 bytes in order of the address currently held, and transmits it to a flash EEPROM. Thus, firmware can perform data transfer by using the header table 133, without distinguishing light data and "involvement elimination" data.

[0217] ECC data are automatically added to 256 bytes each of data transmitted with the NAND bus interface 19, and it is written in the page to which the light block of the flash EEPROM for a light corresponds.

[0218] On the other hand, when digit going up occurs in the value of LWC data, firmware performs the following processings, in order to examine whether swapping actuation is performed.

[0219] That is, firmware leads a MBI table block from the flash EEPROM for a light by performing the 1st read cycle first (step A82). The content of the led MBI table block is stored in the empty area of RAM13.

[0220] Subsequently, firmware searches the led MBI table block, detects the UWC data corresponding to a light block, updates the value of the UWC data, and gets for it blocked and counts it up +one time (step A83). In the MBI table block, as drawing 8 explained, UWC data align in order of a physical block address, and are held. For this reason, firmware can find the UWC data corresponding to a light block easily by using as an index the physical block address which shows a light block.

[0221] The value of the updated UWC data is reflected in a MBI table block and count of rewriting managed table 134 both sides.

[0222] Then, firmware carries out the sequential comparison of the value of the UWC data of the updated light block with the value of the minimum UWC data of the chip for a light, and the value of the minimum UWC data of each of other chip (step A84).

[0223] The value of the minimum UWC data currently held as a value of the minimum UWC data of the chip for a light at the swap order table within the MBI table block led from the chip for a light is used. Without this performing processing in which sorting of the count of rewriting managed table 134 on RAM13 is carried out to the small order of UWC data, although the value of the minimum UWC data of the chip for a light is got to know easily, it can do.

[0224] Moreover, as a value of the minimum UWC data of each of other chip, the value of the count of rewriting managed table 134 on RAM13 is used. By using the value of this count of rewriting managed table 134, the need of carrying out the sequential lead of the MBI table block disappears from each of all other chips other than the chip for a light, and efficient swapping examination processing is realized.

[0225] Subsequently, based on the comparison result of step A84, it investigates whether firmware is more than the threshold (alpha) the difference of the value of the UWC data and the minimum UWC data of the chip for a light with which the light block was updated was beforehand decided to be (step A85), and with [the difference] alpha [more than], swapping actuation within the chip for a light is performed (step A86).

[0226] In this swapping actuation in a chip, firmware leads the block for swapping-ed from the chip for a light first. in this case, the block for swapping-ed -- a minimum of -- it is the block with the value of UWC and the physical block address of this block is specified by the swapping-ed priority NO.1 block number on the swap order table contained in the MBI table block led at step A82. Therefore, without performing sorting of UWC data, firmware can determine the block for swapping-ed easily, and can recognize the physical block address of the block for [the] swapping-ed.

[0227] Next, firmware is between a light block and the block for swapping-ed, and replaces the host address. In this case, the

address translation information on MBI table block ***** led from the address translation table 132 and the chip for a light on RAM132 is updated.

[0228] Next, firmware performs light access for writing the content of the swapping-ed block sucked up on RAM13 for exchange of the content of storage between the light block in the flash EEPROM for a light, and the block for swapping-ed in the light block of the flash EEPROM for a light, and light access for writing the light data for 1 block on RAM13 in the swapping-ed block of a flash EEPROM. Then, firmware performs light access for writing in the content of the MBI table block with which it was updated on RAM13 by the MBI table block in the flash EEPROM for a light for renewal of the MBI table block in the flash EEPROM for a light. These three light accesses are performed with the same gestalt as the case where step A81 explains, respectively.

[0229] Next, processing when the difference of the value of the UWC data and the minimum UWC data of the chip for a light with which the light block was updated in step A86 is under alpha is explained.

[0230] In this case, it investigates whether firmware is more than the threshold (alpha+beta) the difference of the value of the UWC data with which the light block was updated, and the minimum UWC data of one of other chips was beforehand decided to be (step A87), and with [that difference] alpha+beta [more than], swapping actuation between the chip for a light and a chip with the value of that minimum UWC data is performed (step A88).

[0231] In the swapping actuation covering between this chip, firmware leads a MBI table block on RAM13 from a chip with the block for swapping-ed first, and recognizes the swapping-ed priority NO.1 block number on the swap order table contained in that MBI table block.

[0232] Subsequently, firmware leads the block for swapping-ed from the chip for swapping-ed using the physical block address which swapping-ed priority NO.1 block number shows.

[0233] Next, firmware is between a light block and the block for swapping-ed, and replaces the host address. In this case, each address translation information on the address translation table 132 on RAM132, the MBI table block led from the chip for a light, and the MBI table block led from the chip for swapping-ed is updated.

[0234] Next, firmware performs light access for writing the content of the light block in the flash EEPROM for a light, and the swapping-ed block sucked up on RAM13 for exchange of the content of storage during the block for swapping-ed of the chip for swapping-ed in the light block of the flash EEPROM for a light, and light access for writing the light data for 1 block on RAM13 in the swapping-ed block of the flash EEPROM for swapping-ed. Then, firmware performs light access for writing in the content of the MBI table block with which it was updated on RAM13 by the MBI table block in the flash EEPROM for a light, and light access for writing in the content of the MBI table block with which it was updated on RAM13 by the MBI table block in the flash EEPROM for swapping-ed, in order to update the MBI table block of each flash EEPROM for swapping-ed [for a light / flash EEPROM]. These four light accesses are performed with the same gestalt as the case where step A81 explains, respectively.

[0235] Thus, SUWABBINGU actuation is performed between chips within a chip, and, as for the case of the swapping between chips, read access actuation and light access actuation are mostly needed from the case of the swapping in a chip by a unit of 1 time, respectively.

[0236] However, in this example, since the threshold (alpha+beta) for determining whether perform swapping or not is set as the bigger value than the threshold (alpha) in the case of performing swapping actuation in a chip, it can control the occurrence frequency of the swapping between chips compared with the swapping in a chip. Therefore, the degradation by frequent occurrence of the swapping actuation between chips can be prevented.

[0237] When both the differences of the count of replacement are smaller than a predetermined threshold at step A85 and **** A87, firmware performs light access for writing the MBI table block containing the UWC data updated at step A83 in the flash EEPROM for a light (step A89), and, subsequently performs light access processing to a light block. These light access is performed with the same gestalt as the case where step A81 explains, respectively.

[0238] As mentioned above, since light access is immediately performed, without making examination of swapping processing when the digit riser of LWC data does not occur, the time amount of the whole which data writing takes rather than the case where a digit riser occurs can be shortened.

[0239] The value of LWC data is counted up every [1 / +] from the initial value at every light access, and when it amounts to 1K times, a digit riser generates it. For this reason, it is desirable to use not the same value but a discrete value different, respectively as initial value of the LWC data written in all blocks included in a flash EEPROM 11-1 to 11-16 at the time of disk manufacture and shipment. In this case, since the stage when a digit riser occurs shifts for every block even if light access to a contiguous block occurs continuously, the situation of the examination processing and swapping processing of swapping resulting from digit going up concentrating, and generating can be prevented.

[0240] As explained above, in this example, ECC data are generated by the ECC arithmetic circuit 21 for every page, and it is written in the same page as the data stream from which that ECC data was set as the object of an ECC operation. For this reason, even if, even if a two or more pages defect cel occurs in each same bit position, the defect cel contained in the data stream of a flash EEPROM set as the object of an ECC operation is set only to one. Therefore, without using the complicated high ECC operation expression of data bounce-back capability, by the easy usual ECC operation, it becomes possible to perform error detection and correction, and the RAM disk equipment 10 in which rapid access is possible is realized highly [storage dependability].

[0241] Moreover, LWC data are stored in the redundancy field of the head page of each block as count information of rewriting corresponding to the block, and the value of the LWC data is updated in the case of the data writing to the block for

a light. The updated LWC data is added to the light data which should be written in a head page, and is transmitted to the data register 112 of a flash EEPROM. Thus, since LWC data are managed within each block, only by light access to the same block, renewal of data writing and LWC data can be performed, and improvement in the light access engine performance can be aimed at as a result.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] The block diagram showing the configuration of the RAM disk equipment concerning one example of this invention.

[Drawing 2] Drawing showing the structure of the NAND mold flash EEPROM used in the RAM disk equipment of drawing 1.

[Drawing 3] Drawing showing the I/O register group prepared in the RAM disk equipment of drawing 1.

[Drawing 4] Drawing showing an example of the memory map in the RAM disk equipment of drawing 1.

[Drawing 5] Drawing in the flash EEPROM prepared in the RAM disk equipment of drawing 1 showing each two or more blocks utilization gestalt.

[Drawing 6] Drawing showing the data storage format of memory block used for storing of user data in the block shown in drawing 5.

[Drawing 7] Drawing showing the data storage format of the MBI table block used for storing of memory-block management information in the block shown in drawing 5.

[Drawing 8] Drawing showing the content of the address translation stored in the MBI table block of drawing 7, and the UWC table.

[Drawing 9] Drawing showing the content of the swap order table stored in the MBI table block of drawing 7.

[Drawing 10] Drawing showing the data storage format of the KONHIIGU block used for storing of the KONHIIGU information on the RAM disk equipment of drawing 1 in the block shown in drawing 5.

[Drawing 11] Drawing showing the configuration of the address translation table prepared on RAM of the RAM disk equipment of drawing 1.

[Drawing 12] Drawing for explaining the principle of transform processing from the host address to the memory address using the address translation table of drawing 11.

[Drawing 13] Drawing for explaining actuation of the ECC arithmetic circuit established in the RAM disk equipment of drawing 1.

[Drawing 14] The flow chart explaining processing of the firmware in the case of writing data in a flash EEPROM in the RAM disk equipment of drawing 1, and actuation of hardware.

[Drawing 15] The timing chart explaining the program cycle of the flash EEPROM prepared in the RAM disk equipment of drawing 1.

[Drawing 16] The flow chart explaining processing of the firmware in the case of reading data from a flash EEPROM in the RAM disk equipment of drawing 1, and actuation of hardware.

[Drawing 17] The timing chart explaining the sequential read cycle of the flash EEPROM prepared in the RAM disk equipment of drawing 1.

[Drawing 18] The flow chart explaining processing of the firmware in the case of eliminating the data stored in the flash EEPROM in the RAM disk equipment of drawing 1, and actuation of hardware.

[Drawing 19] The timing chart explaining the block erasion cycle of the flash EEPROM prepared in the RAM disk equipment of drawing 1.

[Drawing 20] The flow chart explaining processing of the firmware performed after receiving the light demand from a host system in the RAM disk equipment of drawing 1 before carrying out light access of the flash EEPROM.

[Drawing 21] Drawing showing the data storage format over the flash EEPROM by conventional RAM disk equipment.

[Description of Notations]

10 / -- RAM, 14 / -- A microprocessor, 16 / -- ROM, 17 / -- An external bus interface, 19 / -- A NAND bus interface 21 / -- An ECC arithmetic circuit, 131 / -- A data buffer, 132 / -- An address translation table, 133 / -- A header table, 134 / -- Count of rewriting managed table.] -- RAM disk equipment, 11-1 to 11-16 -- A NAND mold flash EEPROM, 12 -- A controller, 13

[Translation done.]

JPO and NCIP1 are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

[Drawing 1]

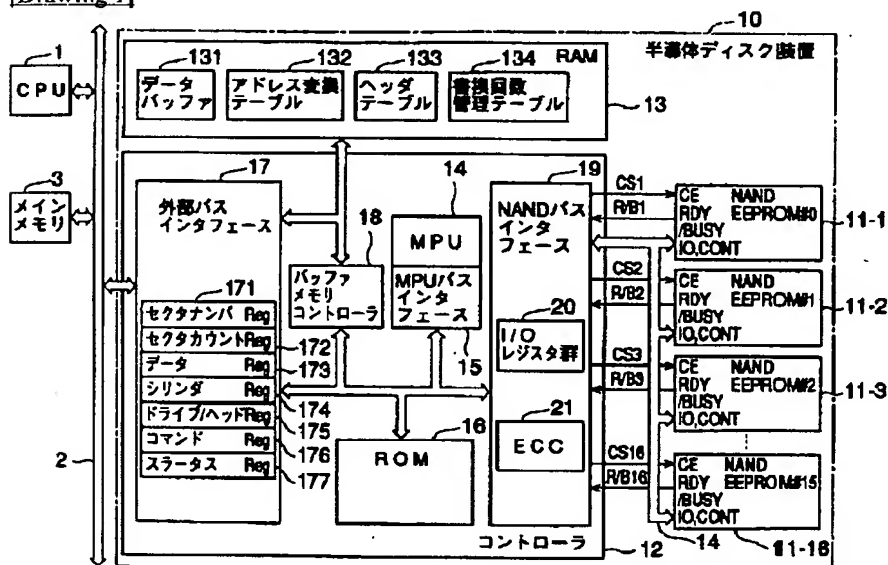
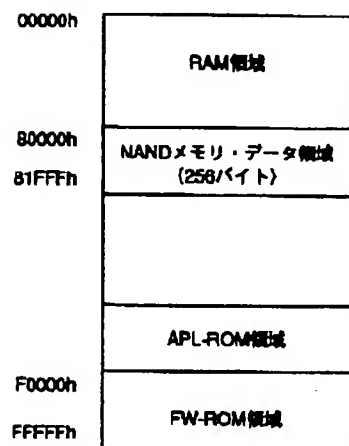


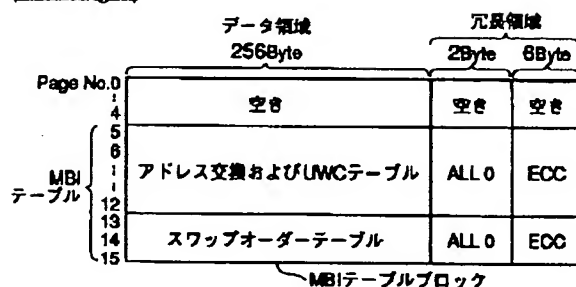
Figure 1: Block diagram of the memory structure of the non-volatile memory. The diagram shows a stack of memory blocks. The top block is labeled "データレジスタ (256+8bit)" and is connected to I/O 7 and I/O 0. Below it is the "メモリアルレイ" (Memory Array). The array is organized into rows (行) and columns (列). The first row is labeled "行1" and the last row is labeled "16". The first column is labeled "256" and the last column is labeled "8". The total size is indicated as "256+8列" and "8K行". The bottom of the array is labeled "0 23 24 25 26 27" and "8bit".

[Drawing 4]

メモリ・マップ



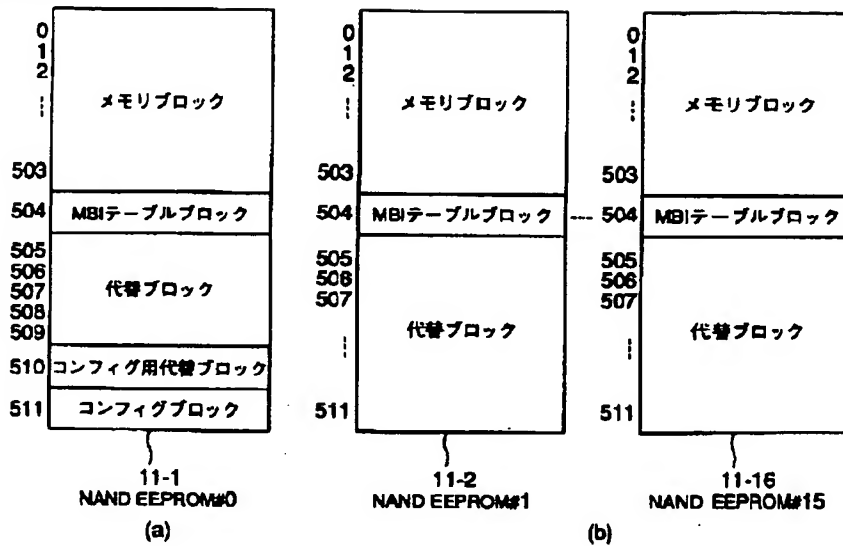
[Drawing 7]



[Drawing 3]

NANDバスインタフェースのI/Oレジスタ群

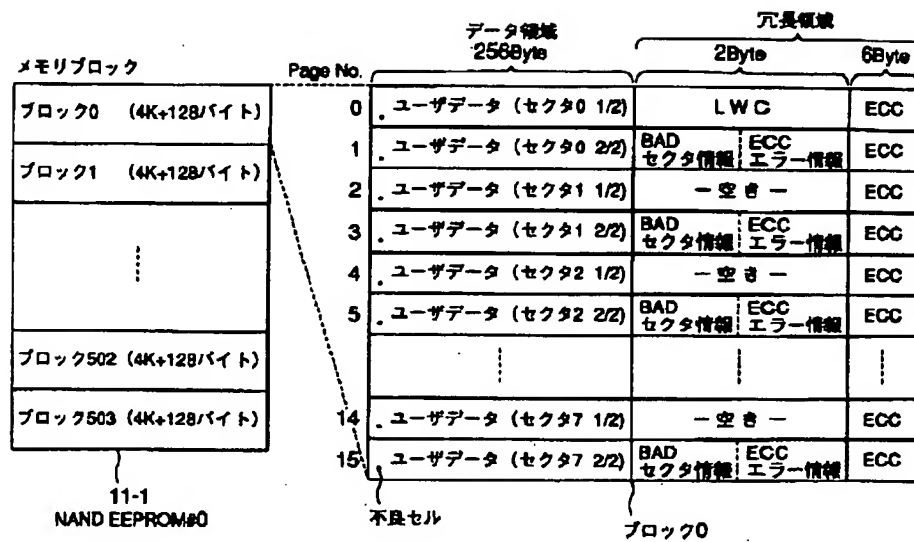
		20
NAND チップNo.	Reg	201
NAND データリード/ライト	Reg	202
NAND CE制御	Reg	203
NAND CLE制御	Reg	204
NAND ALE制御	Reg	205
ECC スタート制御	Reg	206
ECC レディビジーステータス	Reg	207
LWC	Reg	208
ECC データ No.0	Reg	209
ECC データ No.1	Reg	210
ECC データ No.2	Reg	211
ECC データ No.3	Reg	212
ECC データ No.4	Reg	213
ECC データ No.5	Reg	214
NAND チップ No.0 R/B	Reg	215-1
⋮		
NAND チップ No.15 R/B	Reg	215-16

[Drawing 5]
Block No.[Drawing 8]
MBIテーブル内
アドレス

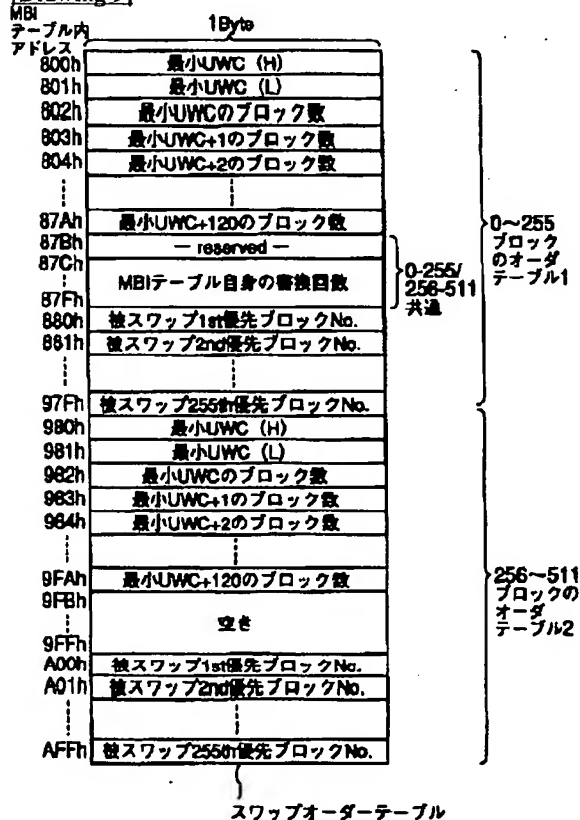
2Byte	
001h,00h	物理ブロック番地0に対応するホスト番地
003h,002h	物理ブロック番地0に対応するUWC
005h,004h	物理ブロック番地1に対応するホスト番地
007h,006h	物理ブロック番地1に対応するUWC
ページ5	
	物理ブロック番地83に対応するホスト番地
	物理ブロック番地83に対応するUWC
ページ6	
ページ11	
	物理ブロック番地448に対応するホスト番地
	物理ブロック番地448に対応するUWC
	物理ブロック番地448に対応するホスト番地
	物理ブロック番地448に対応するUWC
ページ12	
7FDh,7FCh	物理ブロック番地511に対応するホスト番地
7FFh,7FEh	物理ブロック番地511に対応するホスト番地

アドレス変換およびUWCテーブル

[Drawing 6]



[Drawing 9]



[Drawing 11]

ホストアドレス ブロックセクタ アドレス (13bit)	物理アドレス (13bit)	
	チップ番号アドレス (4bit)	チップ内物理ブロック アドレス (9bit)
0		
1		
2		
3		
4		
5		
6		
7		
⋮		
8K		

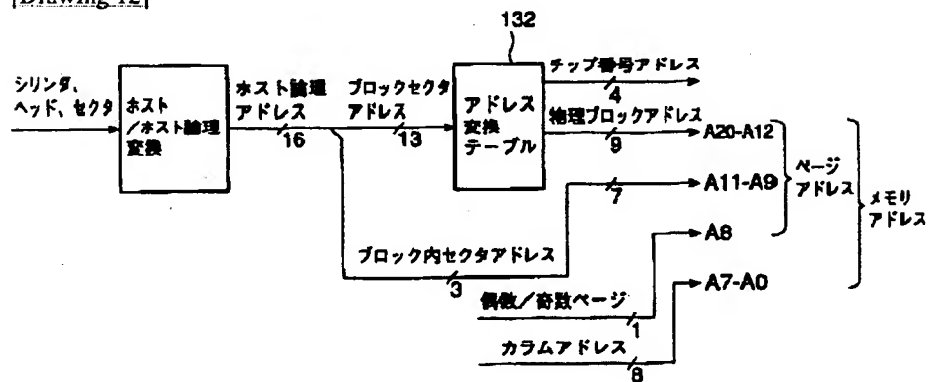
132 アドレス交換テーブル

[Drawing 10]

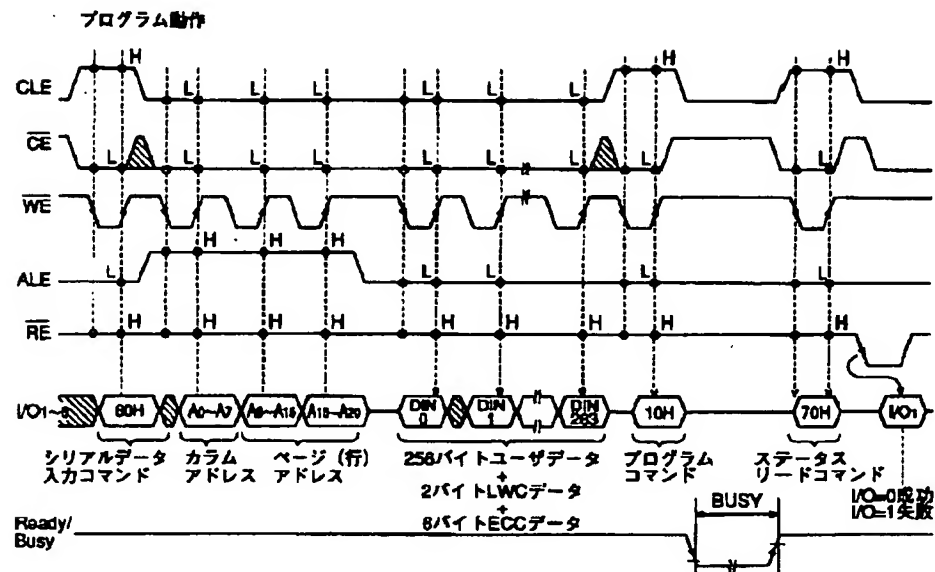
データ領域 256Byte		冗長領域 2Byte 8Byte	
Page No.0	コンヒギ情報	ALL 0	ECC
1	未使用		
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12	CIS情報	ALL 0	ECC
13	ドライブID情報		
14			
15			

コンヒギブロック

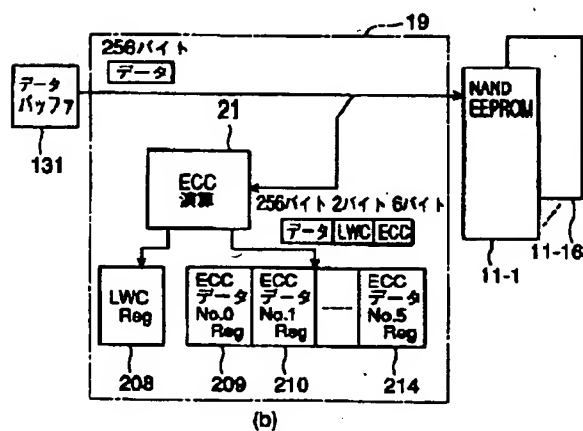
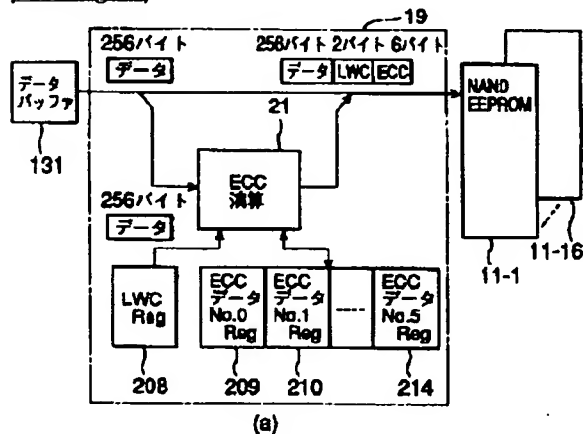
[Drawing 12]



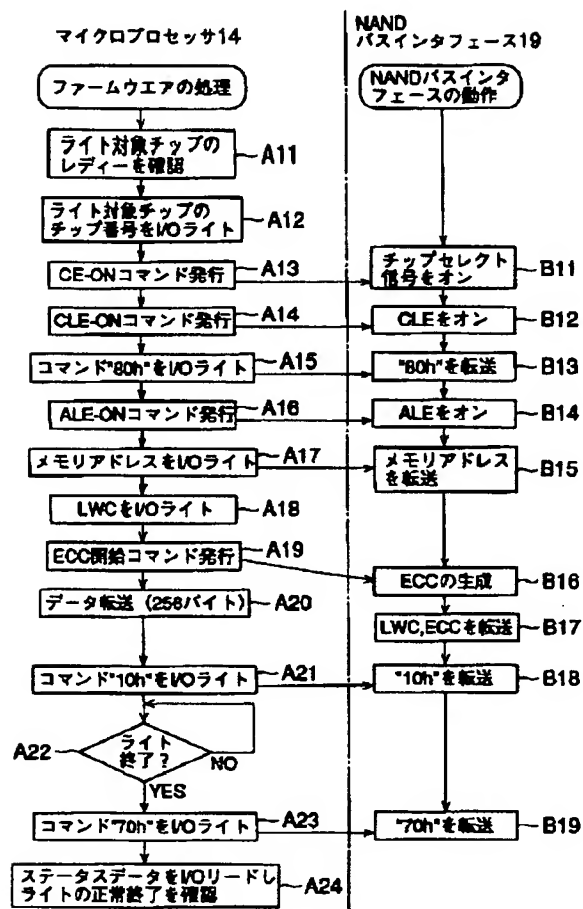
[Drawing 15]



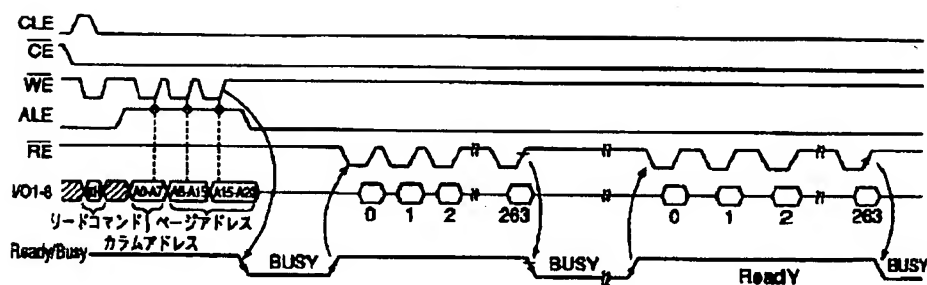
[Drawing 13]



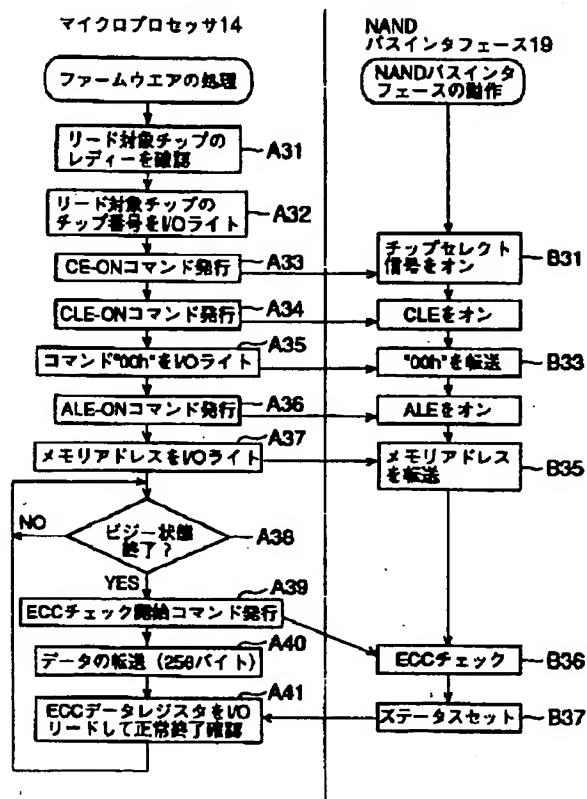
[Drawing 14]



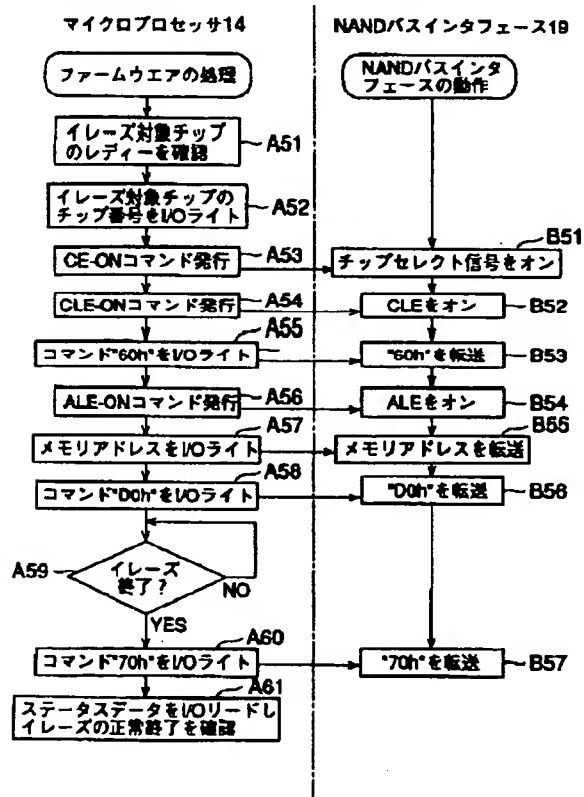
[Drawing 17]
シーケンシャルリード動作

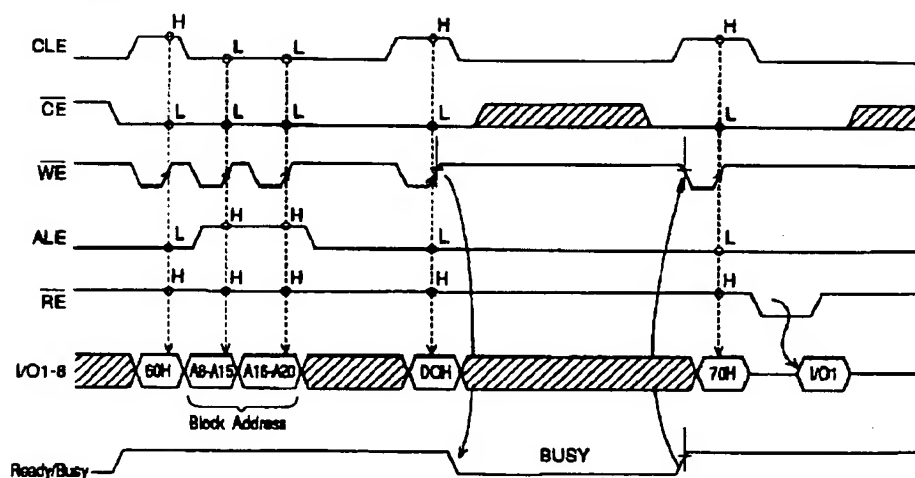


[Drawing 16]

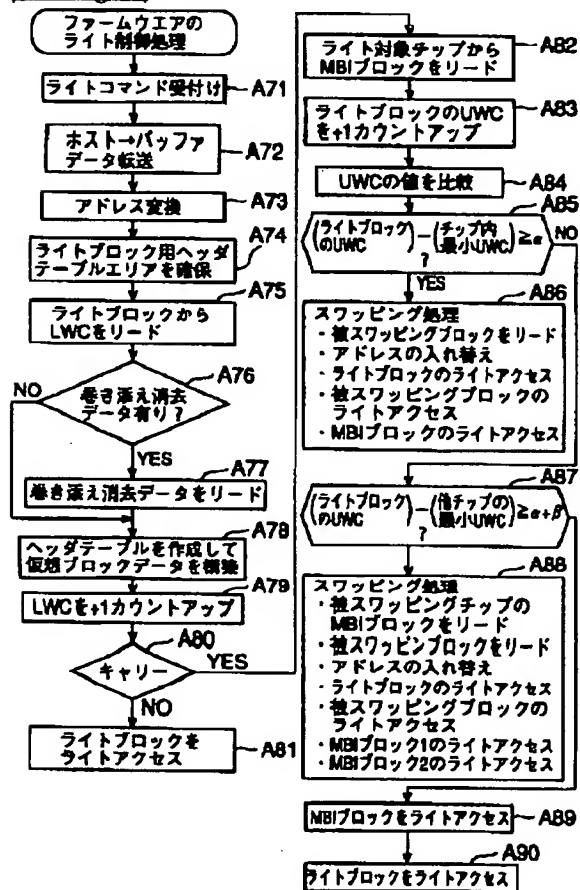


[Drawing 18]

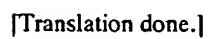


[Drawing 19]
ブロックイレース

[Drawing 20]



[Drawing 21]



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.